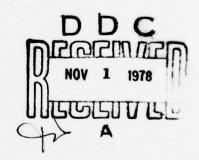
AD-A060 568 TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/6 9/2
A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJECTOR (U)
AUG 78 T C BAKER, R L SIELKEN N00014-78-C-0426 UNCLASSIFIED 1 OF 4 AD A060568 1 Ø - Stadio

TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS 77843



DISTRIBUTION STATEMENT A Approved for public release; Distribution Unlimited

THEMIS 78 10 31 026

 ∞ AD A 0 60 56 TEXAS A&M UNIVERSITY PROJECT THEMIS - TR-Technical Report. A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJECT SCHEDULING PROCEDURE: STATISTICAL PERT by Thomas C. Baker, Jr. Robert L. Sielken Jr. Texas A&M University Office of Naval Research Contract 1:00014-78-C-0426 Project NRO47-179 Reproduction in whole or in part is permitted for any purpose of the United States Government. This document has been approved for public release and sale;

ATTACHMENT I

347380

its distribution is unlimited.

1/8

A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJECT SCHEDULING PROCEDURE: STATISTICAL PERT

by

Thomas C. Baker, Jr. and Robert L. Sielken Jr.

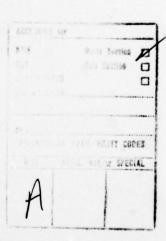
THEMIS OPTIMIZATION RESEARCH PROGRAM
Technical Report No. 57
August 1978

INSTITUTE OF STATISTICS Texas A&M University

Research conducted through the Texas A&M Research Foundation and sponsored by the Office of Naval Research Contract N00014-78-C-0426 Project NR047-179

Reproduction in whole or in part is permitted for any purpose of the United States Government.

This document has been approved for public release and sale; its distribution is unlimited.



ATTACHMENT II

A User's Guide to the Computer Implementation of the New Project Scheduling Procedure: Statistical PERT

by

Thomas C. Baker, Jr. and Robert L. Sielken Jr.

ABSTRACT

This report documents a new project scheduling procedure developed at the Institute of Statistics, Texas A&M University. The project scheduling algorithm is a five step iterative procedure capable of determining a minimum cost project schedule when the activities making up the project have durations which are random variables. The cost of an activity is assumed to be a convex piecewise linear function of the activity's mean duration. The problem is to determine the activity mean durations which both minimize the total project cost and insure that the mean (or some specified percentile) of the corresponding project completion time distribution is less than or equal to a specified project deadline. The entire distribution of the project's completion time under the minimum cost schedule is a valuable by-product. Information on the trade-off between the project's minimum cost and its specified deadline is also provided.

This report is intended as a user's guide to the new project scheduling procedure and its computer implementation. The report includes a description of the project scheduling problem, a general overview of the scheduling procedure including references to technical reports documenting the development of the procedure, and an example of the procedure's performance. The documentation of the computer implementation includes specific input

instructions; sample input and output; flowcharts; individual program descriptions; technical details concerning temporary data sets, job control language, and program interruption and restart procedures; and program listings.

TABLE OF CONTENTS

1.	The Project	Scheduling Problem	1
2.	Outline of S	tatistical PERT	5
3.	An Example o	f Statistical PERT's Performance	7
4.	Concluding R	emarks	16
	References		18
	Appendix A.	Specific Input Instructions for the	
		Statistical PERT Computer System	19
	Appendix B.	Computer Input for the Example in Section 3	29
	Appendix C.	Computer Output for the Example in Section 3	31
	Appendix D.	Flowchart of the Computer System	107
	Appendix E.	Program Descriptions	108
		1. MAIN: Original Input Program	108
		2. DPS: Determinisitc Project Scheduling	109
		3. DSR: Deterministic Schedule Resolution	110
		4. SIMP: Simplification	111
		5. MODSIMP: Subsequent Simplification	115
		6. SUBNET: Subnetwork Analysis	116
		7. DECOMP: Decomposition	136
		8. SYNTH: Synthesis	139
	Appendix F.	Structure and Contents of the Temporary	
		Data Sets	142
		1. FILEO	143
		2. FILE1	145
		3. FILE2	147
		4. FILE3	148

Appendix F.	(Continued)	
	5. FILE4	150
	6. FILE5	151
	7. FILE6	153
	8. FILE7	154
Appendix G.	Job Control Language for Texas A&M University	
	Computer Facilities	157
	1. Execution using FORTRAN source decks	157
	2. Execution using load modules	160
Appendix H.	Interrupting the Computer System	162
	1. SAVEFIL: System Interruption Program	164
	2. READFIL: System Restart Program	165
	3. JCL for SAVEFIL and READFIL	166
Appendix I.	Program Listings	168
	1. MAIN	169
	2. DPS	175
	3. DSR	187
	4. SIMP	192
	5. MODSIMP	200
	6. DECOMP	227
	7. SUBNET	242

1. The Project Scheduling Problem

A project is composed of several "tasks" or "activities." These activities can be represented by arcs in a directed network. For example, a small project might consist of activities A, B, C, D, and E with the following precedence relationships:

- (i) A must be completed before either C or D can be begun;
- (ii) B must be completed before D can be begun; and
- (iii) C and D must both be completed before E can be begun. The corresponding network representation is shown in Figure 1. The arc labeled F does not correspond to any "real" activity but is a "dummy" activity merely representing the precedence relation that A must be completed before D can be begun. The points numbered 1, 2, ..., 5 are called nodes. In the network representation of a project the activities originating at a node can be begun only after all activities terminating at that node have been completed.

The time that it actually takes to complete an activity once that activity has been begun is called the activity's duration and is a random variable. The cost of an activity is assumed to be a convex piecewise linear function of the activity's mean duration time. An example of an activity's cost curve is given in Figure 2. In this example TIME(1) is the minimum mean duration time that can be scheduled. TIME(4) is the cheapest mean duration and hence the maximum mean duration that would be scheduled. Of course a linear cost curve is the simplest convex piecewise linear cost function. The more general piecewise behavior, however, frequently arises if there are alternative methods of performing an activity. These methods do not differ in the end result but do differ in the amount of time they take and their cost. For example, to have a mean

duration in the interval [TIME(1), TIME(2)] might require the use of a very expensive piece of special equipment while having a mean duration in the interval [TIME(2), TIME(3)] requires only specially trained personnel and having a mean duration in the interval [TIME(3), TIME(4)] just requires varying amounts of standard resources. The form of the activity duration distribution may vary from one time interval to another. For example, the activity duration distribution might be a beta distribution when the mean duration is in [TIME(1), TIME(3)] and approximately a normal distribution when the mean duration is in [TIME(3), TIME(4)].

A project schedule is a specification of each activity's mean duration. The total project cost is simply the sum of the corresponding activity costs. The time to complete the entire project is a random variable whose distribution depends upon the activity duration distributions. The objective is to determine a minimum cost project schedule such that the mean of the corresponding project completion time distribution (or some specified percentile of the project completion time distribution) is less than or equal to a specified project deadline.

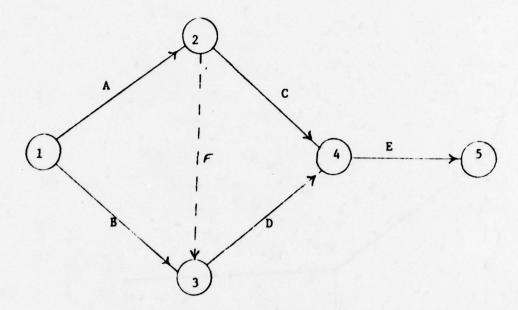


Figure 1. A Small Project Represented as a Directed Network

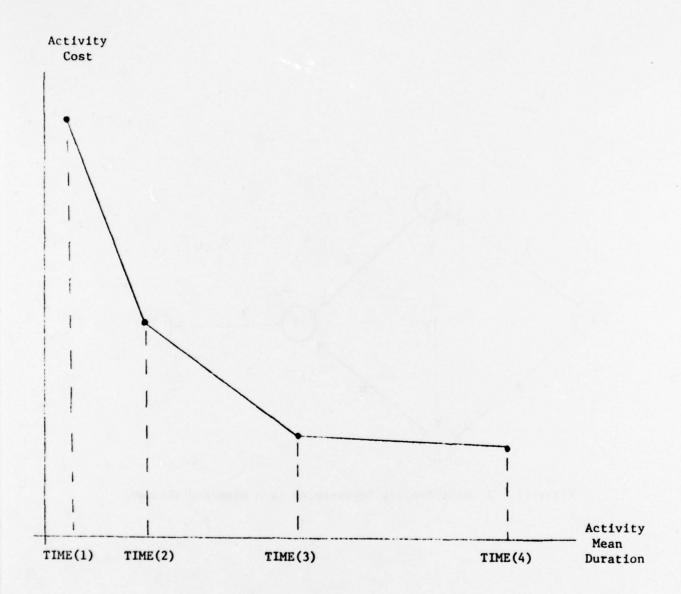


Figure 2. An Activity's Cost as a Function of the Activity's Mean Duration

2. Outline of Statistical PERT

In 1974 the development of a new project scheduling procedure was begun with the support of the Office of Naval Research. The new project scheduling procedure that has resulted is an iterative algorithm involving the following five general steps:

- Step 1. Deterministic Scheduling: Find a minimum cost project schedule which completes the project by TARGET TIME when each activity's duration is exactly its mean duration and hence deterministic instead of random. (The initial value of TARGET TIME is usally the specified project deadline.)
- Step 2. Simplification: Let each activity's duration be a random variable with distribution corresponding to that activity's mean duration chosen during Deterministic Scheduling.

 Replace various configurations of activities by single activities. The duration distribution for a replacement activity is the distribution of the time to complete all of the activities in the configuration it is replacing. The result of this step is a simplified project network with fewer activities.
- Step 3. Decomposition: Partition the simplified project network into several subnetworks in such a way that the resultant subnetworks can be linked together in either series or parallel to form the simplified project network.
- Step 4. Subnetwork Analysis: Analyze separately each of the subnetworks determined during Decomposition. Within a subnetwork each activity's duration distribution is approximated by a two-point discrete distribution with

matching mean, variance, and third moment. Determine the subnetwork duration distribution corresponding to these discrete activity duration distributions.

Step 5. Synthesis: Combine the approximate subnetwork duration distributions to obtain an approximate completion time distribution for the entire project. If the mean (or some specified percentile), T, of this project completion time distribution is sufficiently close to the specified project deadline, the "optimal" project schedule has been found. Otherwise, reset TARGET TIME to New TARGET TIME = Old TARGET TIME* (Project Deadline/T) and return to Step 1.

A general discussion and relatively nonmathematical overview of this project scheduling procedure is contained in Appendix D, "Flowchart of the Computer System," and Appendix E, "Program Descriptions" (see also Sielken and Hartley (1977)). The detailed documentation of the development thus far of each step is as follows:

- Step 1. Dunn and Sielken (1977);
- Step 2. Hartley and Wortham (1966) and Ringer (1969);
- Step 3. Sielken and Fisher (1976);
- Step 4. Sielken, Ringer, Hartley, and Arseven (1974), Sielken, Hartley, and Spoeri (1976), and Baker and Sielken (1978);
- Step 5. Sielken, Ringer, Hartley and Arseven (1974) and Sielken, Hartley, and Spoeri (1976).

The forthcoming Technical Report No. 60, "Statistical PERT: The Precision of the Estimated Project Completion Time Distribution," will also be of interest to the user.

3. An Example of Statistical PERT's Performance

A small project network is depicted in Figure 3. The relationship between each activity's mean duration and its cost is given in Table 1. The project scheduling algorithm also requires that the activity's duration distribution be specified at the midpoint of each time interval on the convex piecewise linear cost function, i.e., when the activity's mean duration is [TIME(1) + TIME(2)]/2, [TIME(2) + TIME(3)]/2, etc. This information is also given in Table 1. The algorithm assumes that if an activity's mean duration is not at the midpoint but at c times the midpoint and still in the same time interval, then the activity's duration distribution has the same form (Normal, Beta, Constant, etc.) but with a new variance equal to c² times the variance at the midpoint. Thus, for example, if activity A's mean duration is 28, its cost is 34, and activity A's duration distribution is Beta [10,40] with mean 28 and variance (28/25)²36.

With a project deadline of 125 the algorithm's iterative determination of the minimum cost project schedule is as follows:

Step 1. Deterministic Scheduling: The shortest feasible project completion time when each activity duration is its mean duration is found to be 90 by determining the longest path through the project network when each activity's duration is equal to its minimum mean duration, TIME(1). Similarly, the longest such feasible project completion time is found to be 135 by determining the longest path through the project network when each activity's duration is equal to its maximum mean duration. The minimum cost schedule which completes the project by TARGET TIME when each activity's duration is its mean duration is determined for each value of TARGET TIME

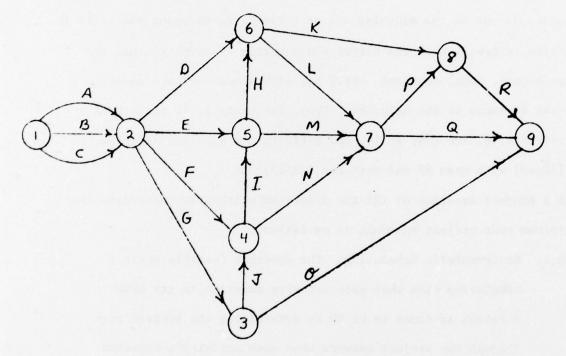


Figure 3. Example Project Network

Table 1. Activity Costs and Duration Distributions

Activity	Mean Durations	Cost	Duration Distribution at Midpoint
A	TIME(1) = 10	100	
	TIME(2) = 15	70	Beta on [5,20] with mean 12.5 and variance 16
	TIME(3) = 20	50	Normal with mean 17.5 and variance 25
	TIME(4) = 30	30	Beta on [10,40] with mean 25 and variance 36
В	TIME(1) = 8	60	
	TIME(2) = 14	45	Normal with mean 11 and variance 9
С	TIME(1) = 10	50	Constant Duration = 10
D	TIME(1) = 50	200	
	TIME(2) = 70	160	Normal with mean 60 and variance 100
E	TIME(1) = 32	45	
	TIME(2) = 40	40	Normal with mean 36 and variance 18
F	TIME(1) = 20	64	
	TIME(2) = 32	48	Beta on [10,50] with mean 26 and variance 70
G	TIME(1) = 13	30	
	TIME(2) = 19	25	Normal with mean 16 and variance 7
н	TIME(1) = 20	60	
	TIME(2) = 30	50	Normal with mean 25 and variance 12
	TIME(3) = 35	48	Normal with mean 32.5 and variance 9
1	TIME(1) = 5	60	Constant Duration = 5
J	TIME(1) = 18	62	
	TIME(2) = 26	49	Beta on [10,40] with mean 22 and variance 6
K	TIME(2) = 20	40	Constant Duration = 10
			Constant Dutation - 10
L	TIME(1) = 6	75	Normal with mean 8 and variance 5
	TIME(2) = 10	50	

Table 1. (Continued)

Activity	Mean Durations	Cost	Duration Distribution at Midpoint
м	TIME(1) = 36	225	N1
	TIME(2) = 44	175	Normal with mean 40 and variance 80
N	TIME(1) = 30	400	Name 1 and the same 25 and condense 25
	TIME(2) = 40	300	Normal with mean 35 and variance 25
0	TIME(1) = 60	250	Normal with mean 70 and variance 144
	TIME(2) = 80	210	Normal with mean 70 and variance 144
P	TIME(1) = 4	52	Constant Duration = 4
Q	TIME(1) = 2	100	Beta on [1,6] with mean 3 and variance 1
	TIME(2) = 4	90	beta on [1,0] with mean 3 and variance 1
R	TIME(1) = 4	110	Normal with mean 5 and variance 2
	TIME(2) = 6	80	Morman with mean 2 and variance 2

between 90 and 135. The corresponding optimal activity mean durations are given in Table 2. The project cost curve is depicted in Figure 4.

The initial activity mean durations are those corresponding to TARGET TIME = 125; namely,

$$A = 30, F = 32, K = 10, P = 4,$$

$$B = 14$$
, $G = 13$, $L = 10$, $Q = 4$,

$$C = 10$$
, $H = 34$, $M = 44$, $R = 6$.

$$D = 70, I = 5, N = 40,$$

$$E = 40, J = 23, O = 80,$$

Step 2. Simplification: The only configuration of activities which can be readily replaced by an equivalent single activity is A, B, and C in parallel. If the replacement activity is denoted by ABC, then the simplified project network consists of the single activity ABC and the original activities D, E, ..., R. The initial duration distribution for ABC as a function of t is

$$F_{ABC}(t) = F_A(t)F_B(t)F_C(t)$$

where $F_A(t)$ denotes a beta distribution with mean 30 and variance $(30/25)^2[(40-30)^2/36]$, $F_B(t)$ denotes a normal distribution with mean 14 and variance $(14/11)^29$, and $F_C(t)$ denotes the distribution for a constant duration of 10.

Step 3. Decomposition: The simplified project network is partitioned into two subnetworks in series. The first subnetwork, SUB₁, consists of the single activity ABC. The second subnetwork, SUB₂, consists of the activities D, E, ..., R.

Table 2. Optimal Activity Mean Durations for All Feasible TARGET TIME's

TARGET TIME		134-D	128-D	124-D	114-D	113-D	108-D	106-D	105-D	104-D	100-D	0-96	92-D
Range of D	0<0<1	0 <d<6< th=""><th>0<d<4< th=""><th>0<d<10< th=""><th>0<d<1< th=""><th>0<d<5< th=""><th>0<d<2< th=""><th>0<d<1< th=""><th>0<d<1< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<></th></d<4<></th></d<1<></th></d<1<></th></d<2<></th></d<5<></th></d<1<></th></d<10<></th></d<4<></th></d<6<>	0 <d<4< th=""><th>0<d<10< th=""><th>0<d<1< th=""><th>0<d<5< th=""><th>0<d<2< th=""><th>0<d<1< th=""><th>0<d<1< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<></th></d<4<></th></d<1<></th></d<1<></th></d<2<></th></d<5<></th></d<1<></th></d<10<></th></d<4<>	0 <d<10< th=""><th>0<d<1< th=""><th>0<d<5< th=""><th>0<d<2< th=""><th>0<d<1< th=""><th>0<d<1< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<></th></d<4<></th></d<1<></th></d<1<></th></d<2<></th></d<5<></th></d<1<></th></d<10<>	0 <d<1< th=""><th>0<d<5< th=""><th>0<d<2< th=""><th>0<d<1< th=""><th>0<d<1< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<></th></d<4<></th></d<1<></th></d<1<></th></d<2<></th></d<5<></th></d<1<>	0 <d<5< th=""><th>0<d<2< th=""><th>0<d<1< th=""><th>0<d<1< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<></th></d<4<></th></d<1<></th></d<1<></th></d<2<></th></d<5<>	0 <d<2< th=""><th>0<d<1< th=""><th>0<d<1< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<></th></d<4<></th></d<1<></th></d<1<></th></d<2<>	0 <d<1< th=""><th>0<d<1< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<></th></d<4<></th></d<1<></th></d<1<>	0 <d<1< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<></th></d<4<></th></d<1<>	0 <d<4< th=""><th>0<d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<></th></d<4<>	0 <d<4< th=""><th>0<d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<></th></d<4<>	0 <d<4< th=""><th>0<d<2< th=""></d<2<></th></d<4<>	0 <d<2< th=""></d<2<>
Project Cost When D=0	1552.0	1552.4	1557.4	1563.9	1583.9	1586.1	1606.1	1614.6	1620.2	1626.2	1660.2	1702.8	1747.8
Activity	,												
A	30	30	30	30-D	20	20-D	15	15	15-D	14-D	10	. 01	10
8	14	14	14	14	14	14	14	14	14	14-D	10	10	10
J	9	10	10	10	10	10	10	10	10	10	10	10	10
0	20	20	20	70	20	20	20	20	70	70	70-D	0-99	62
ы	07	04	07	07	d-04	39	39-D	37-D	36	36	36	36	36
4	32	32	32	32	32	32	32	32-D	31	31	31	31	31
5	19	19-D	13	13	13	13	13	13	13	13	13	13	13
#:	35-D	34	34	34	34	34	34	34	34	34	34-D	30-D	26
1	5	2	2	S	2	2	S	2	2	2	2	5	2
7	56	56	26-D	22	22-D	21	21-D	19-D	18	18	18	18	18
×	9	10	10	10	10	10	9	10	10	10	10	10	10
1	10	10	10	10	10	10	10	10	10	10	10	10	10
×	77	77	77	77	77	77	77	77	77	77	44-D	40-D	36
N	40	40	07	07	40	07	07	07	07	07	07	70	70
0	80	80	80	80	80	80	80-D	78-D	77	77	77-D	73-D	0-69
Ь	4	4	4	4	4	7	4	4	7	7	4	4	4
0	4	4	4	4	4	7	4	4	4	4	4	4	4
R	9	9	9	9	9	9	9	9	9	9	9	9	Q-9

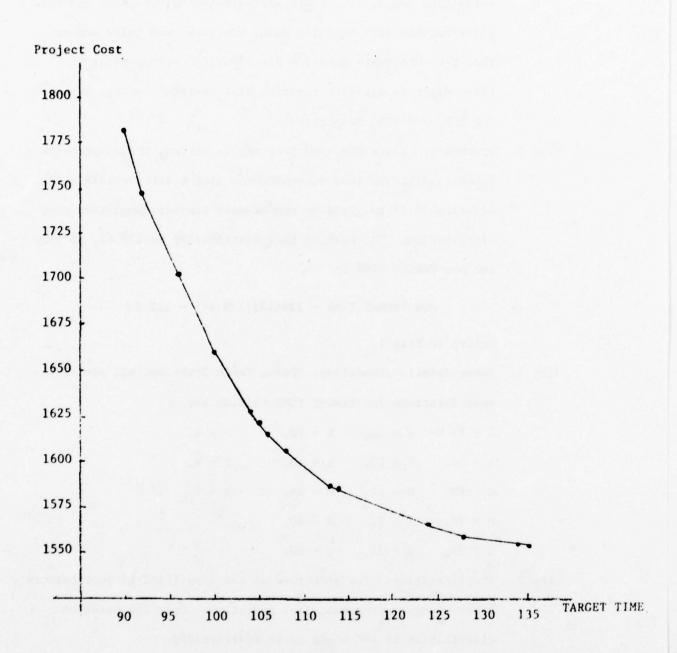


Figure 4. Total Project Cost as a Function of TARGET TIME

- Step 4. Subnetwork Analysis: The activity duration distributions for activities ABC,D, ..., R are approximated by two-point discrete distributions with matching mean, variance, and third moment.

 Then the subnetwork duration distribution corresponding to these discrete activity duration distributions is determined for SUB₁ and SUB₂ respectively.
- Step 5. Synthesis: Since SUB₁ and SUB₂ are in series, their approximate duration distributions determined in Step 4 are combined using equation (8.1) to yield an approximate project completion time distribution. The mean of this distribution is 139.42, so that the new TARGET TIME is

New TARGET TIME = 125(125/139.42) = 112.07

Return to Step 1.

Step 1. Deterministic Scheduling: Using Table 2 the optimal activity

mean durations for TARGET TIME = 112.09 are

A = 19.07 F = 32, K = 10, P = 4,

B = 14, C = 13, L = 10, Q = 4,

C = 10, H = 34, M = 44, R = 6,

D = 70, I = 5, N = 40,

E = 39, J = 21, O = 80,

- Step 2. Simplification: The structure of the simplified project network never changes after the first iteration. Only the duration distribution of ABC needs to be redetermined.
- Step 3. Decomposition: The partitioning of the simplified project network never changes after the first iteration.

- Step 4. Subnetwork Analysis: The new activity duration distributions are approximated by new two-point discrete distributions. Then new approximate duration distributions for ${\tt SUB}_1$ and ${\tt SUB}_2$ are determined.
- Step 5. Synthesis: The new approximate project mean completion time is 128.42, so that the new TARGET TIME is

New TARGET TIME = 112.07(125/128.42) = 109.09.

Return to Step 1.

If 128.42 was sufficiently close to the specified project deadline of 125, then these optimal activity mean durations would constitute the minimum cost project schedule. Otherwise, the project scheduling algorithm could be continued.

4. Concluding Remarks

The new project scheduling procedure allows the project scheduler to specify

- (i) the precedences among the project's activities,
- (ii) the relationship between an activity's cost and its mean duration,
- (iii) the manner in which an activity's actual duration varies about its mean duration, and
- (iv) a deadline for either the project's mean completion time or a prescribed percentile of the project completion time distribution.
 In return the project scheduler receives
 - (i) a minimum cost project schedule which delineates each activity's mean duration time,
 - (ii) an estimate of the distribution of the project completion time,
 - (iii) information on the trade-off between the project's minimum cost and its specified deadline, and
 - (iv) a tool for monitoring the project's progress and, if need be, rescheduling.

An exciting feature of this new project scheduling procedure is that it simultaneously incorporates the desire to minimize the project cost and the realization that an activity's duration is not necessarily a fixed quantity exactly equal to its prescribed duration but rather a random quantity varying about a prescribed duration. No longer must the project scheduler either (i) choose a reasonable cost schedule which heuristically hedges against the randomness in the activities he guesses will be critical, or (ii) choose a reasonable schedule which should probably finish before the deadline and then guess where he can save money without disturbing the suspected completion time too much. By considering both cost and randomness

together in one systematic algorithm, the new project scheduling procedure eliminates this guesswork.

The authors wish to acknowledge their gratitude to the Office of Naval Research for the support of this research under contracts N00014-68-A-0140 and N00014-76-C-0038. Several present and past members of the Institute of Statistics at Texas A&M University have also contributed to the development of this new project scheduling procedure and its computer implementation:

E. Arseven (Lederle Laboratories), P. P. Biemer, C. S. Dunn, N. E. Fisher (Compucon Inc.), L. J. Ringer, and R. K. Spoeri (Bureau of the Census).

The authors also want to particularly acknowledge the considerable contributions of H. O. Hartley to Statistical PERT.

References

- Baker, Thomas C. Jr. and R. L. Sielken Jr., "Statistical PERT: Improvements in the Determination of the Project Completion Time Distribution", THEMIS Technical Report 58, August 1978.
- Baker, Thomas C. Jr. and R. L. Sielken Jr., "Statistical PERT: The Precision of the Estimated Project Completion Time Distribution", THEMIS Project Technical Report No. 60, August 1978.
- Biemer, Paul P., and R. L. Sielken Jr., "Incorporating Project Cost Considerations into Stochastic PERT", THEMIS Project Technical Report No. 52, November, 1975.
- Dunn, C. S. and R. L. Sielken Jr., "Statistical PERT: An Improved Project Scheduling Algorithm", THEMIS Project Technical Report No. 55, February, 1977.
- Fulkerson, D. R., "A Network Flow Computation for Project Cost Curves", Management Science, Vol. 7, No. 2, (January, 1961), pp. 167-178.
- Hartley, H. O. and A. W. Wortham, "A Statistical Theory for PERT Critical Path Analysis," Management Science, Vol. 12, No. 10, (June, 1966), pp. 469-481.
- Ringer, L. J., "Numerical Operators for Statistical PERT Critical Path Analysis", Management Science, Vol. 16, No. 2, (October, 1969), pp. B-136 B-143.
- Sielken, R. L. Jr. and Norman E. Fisher, "Statistical PERT: Decomposing a Project Network", THEMIS Project Technical Report No. 50, January, 1976.
- Sielken, R. L. Jr. and H. O. Hartley, "A New Statistical Approach to Project Scheduling", THEMIS Project Technical Report No. 56, December 1977.
- Sielken, R. L. Jr., H. O. Hartley and R. K. Spoeri, "Statistical PERT: An Improved Subnetwork Analysis Procedure", THEMIS Project Technical Report No. 51, January 1976.
- Sielken, R. L. Jr., L. J. Ringer, H. O. Hartley and E. Arseven, "Statistical Critical Path Analysis in Acyclic Stochastic Networks: Statistical PERT", THEMIS Project Technical Report No. 48, November, 1974.
- Spoeri, R. K., L. J. Ringer and R. L. Sielken Jr., "A Statistical Procedure for Optimization of PERT Network Scheduling Systems", THEMIS Project Technical Report No. 53, April, 1976.

Appendix A

Specific Input Instructions for the Statistical PERT Computer System

Only the main program (MAIN) requires any input data cards. The order and setup of those cards is indicated below.

Card			
Order	Column	Format	Description
1	1-10	110	NACT = the number of activities in the
			project network: NACT < 1000.
	11-20	110	NODES = the number of nodes in the
			project network; NODES < 1000.
	21-30	110	NSRCE = the number of the project's
			source node; 1 < NSRCE < 9999.
	31-40	110	NSINK = the number of the project's
			sink node; 1 ≤ NSINK ≤ 9999.
	41-50	110	LNODEN = the largest node number in
			the project network; LNODEN < 9999.

Repeat cards 2 - 5 for each activity in the project network.

2	1-10	110	I = the number of the activity being
			described; $1 \le I \le 9999$. (Note: The
			network's activities need not be
			numbered consecutively, and they
			need not be described in order.)
	11-20	110	NODEO = the activity's origin node
			number; $1 \leq \text{NODEO} \leq 9999$.
	21-30	110	NODET = the activity's terminal node
			number; 1 < NODET < 9999.

	31-40	110	NCT = the number of completion times
			and costs needed to specify this
			activity's piecewise linear cost curve
			as a function of its mean duration;
			2 ≤ NCT ≤ 6. See Figure A.1 for
			clarification. (Note: For activities
			with constant or fixed duration, NCT \equiv 2.)
3	1-10	110	TIME(1).
	11-20	110	TIME(2).
(NCT-1)*	10+1-NCT*10	110	TIME (NCT)
			TIME(J) = the J-th activity completion
			time; $0 \le \text{TIME}(J) < 32768 = 2^{15}$. These
			times must be in increasing order. See
			Figure A.1 for clarification. (Note:
			For activities with constant duration
			$TIME(1) \equiv TIME(2).)$
4	1-10	110	COST(1).
	11-20	110	COST(2).
(NCT-1)*	10+1-NCT*10	110	COST (NCT).
			COST(J) = the cost associated with the
			J-th activity completion time;
			$0 \le COST(J) < 32768 = 2^{15}$. See
			Figure A.1 for clarification. (Note:
			For activities with constant duration,
			COST(1) = COST(2).)

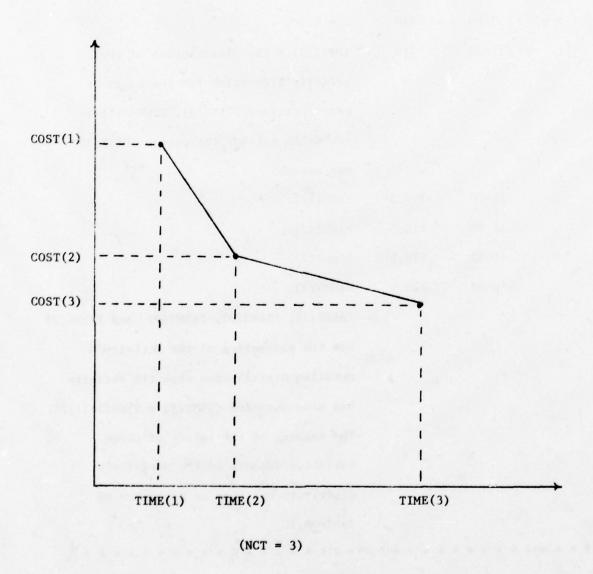


Figure A.1

An example of an activity's cost curve.

Card 5 is a "packet" composed of NCT - 1 cards each containing the following information.

5	1-10	110	J = 1, 2,, NCT - 1.
	11-20	110	<pre>IDIST(J) = the distribution of the</pre>
			activity's duration for the range of
			<pre>mean durations (TIME(J), TIME(J+1));</pre>
			see Table A.1 for the values IDIST(J)
			may assume.

21-30 F10.5 PARM1(J). 31-40 F10.5 PARM2(J).

41-50 F10.5 PARM3(J).

50-60 F10.5 PARM4(J).

PARM1(J), PARM2(J), PARM3(J), and PARM4(J) are the parameters of the activity's duration distribution when the activity has mean duration (TIME(J) + TIME(J+1))/2. The meaning of the values of these variables depends on the activity's distributional type as described in Table A.1.

6 1-10 I10 TEST1.
11-20 I10 TEST2.

TEST1 and TEST2 control the amount of output from the deterministic project scheduling program (DPS).

Table A.1

Activity Duration Distributions and Their Parameters

Distribution	Bet	ta	Nor	ma1	Rectangular	Dummy
IDIST	-	1		0	1	2
PARM1	Min Time	Min Time	Min Time		Min Time	Fixed Time
PARM2	Mode	Alpha, α	Mean, µ	Mean, μ		
PARM3	Max Time	Max Time	Max Time	St Dev., σ	Max Time	
PARM4	-1	Beta, β	-1	1		
	а	b	С	d	e	f

Interpreted Distribution:

a)
$$f(x) = \frac{\Gamma(a+b+2)(x-Min)^a(Max-x)^b}{\Gamma(a+1)\Gamma(b+1)(Max-Min)^{a+b+1}}$$
 Min $\leq x \leq Max$ where a, b chosen so that $\mu = \frac{Min+4*Mode+Max}{6}$ and $\sigma^2 = \frac{(Max-Min)^2}{36}$

b)
$$f(x) = \frac{\Gamma(\alpha+\beta+2)(x-Min)^{\alpha}(Max-x)^{\beta}}{\Gamma(\alpha+1)\Gamma(\beta+1)(Max-Min)^{\alpha+\beta+1}}$$
, $Min \le x \le Max$, $\alpha > -1$ $\beta > -1$

which implies
$$\mu = \frac{Min(\beta+1)+Max(\alpha+1)}{(\alpha+\beta+2)}$$

c)
$$f(x) = \frac{1}{\sqrt{2\pi} s} e^{-\frac{1}{2}(\frac{x-\mu}{s})^2}$$
,

where s = (Max - Min)/6 and $\mu \ge 3s$ is reasonable

d)
$$f(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$
,

where $\mu \ge 3\sigma$ is reasonable

e)
$$f(x) = \frac{1}{(Max-Min)}$$
, $Min \le x \le Max$

f)
$$f(x) = 1$$
, $x = FIXED TIME$

TEST1 = input data option.

TEST1 = 0 implies all the
 information input to the
 program is to be printed.

TEST1 = 1 implies the input data

is not to be printed.

TEST2 = intermediate output option.

TEST2 = 0 implies the output

arising from intermediate

calculations is to be

printed in addition to

the optimal activity mean

durations for every

feasible completion time.

TEST2 = 1 implies the optimal
activity mean durations
for every feasible project
completion time are to be
printed but the intermediate calculations are
not to be printed.

21-30 IIO JMAT = an output option in the simplification program (SIMP).

for each activity in the simplified network is to be printed (minimum output).

JMAT = 0 implies the minimum

output plus a listing of

which activities were

simplified and what

activity configurations

are to be printed (com
plete output).

The quantities on card 7 are all parameters used by the subnetwork analysis program (SUBNET). See Appendix D for a more detailed description of Subnetwork Analysis.

7	1-10	110	<pre>IEDF = the number of subdivisions in the</pre>
			estimated duration distribution for each
			subnetwork; 2 < IEDF < 20.
	11-20	110	NMAX = the maximum number of activities
			in a subnetwork for which the subnetwork's
			entire discrete duration distribution is
			to be automatically enumerated; since
			2 ^{NMAX} calculations are required, a reason-
			able value for NMAX is ≤ 12 .
	21-30	110	IPOOL = the bound computation option.
			IPOOL = 0 implies the maximum-
			cluster procedure is to
			be used to determine the

upper and lower bounds (F_2^+, F_2^-) on each subnetwork's discrete duration distribution.

IPOOL = 1 implies the unioncluster procedure is to
be used to determine the
upper and lower bounds $(F_1^+, F_2^-) \text{ on each subnet-}$ work's discrete duration
distribution.

31-40 IIO SAMSIZ = the maximum number of activity duration configurations to be explicitly considered in the determination of the upper and lower bounds on each subnetwork's discrete duration distribution; SAMSIZ \(\geq \) 2NMAX.

Card 8 contains additional parameters used by the subnetwork analysis program (SUBNET).

8 1-10 II0 THELAM = (θ, λ) pair option.

THELAM = 0 implies the standard (θ,λ) pairs are to be used when determining clusters, i.e., $(\theta,\lambda)=(1,1),\ (2,2),\ (3,2).$

THELAM = 1 implies the three specified (θ, λ) pairs are to be used when determining clusters instead of the standard pairs.

11-20	F10.5	THETA(1).
21-30	F10.5	THETA(2).
31-40	F10.5	THETA(3).
41-50	F10.5	LAMBDA(1).
51-60	F10.5	LAMBDA(2).
61-70	F10.5	LAMBDA(3).
		murm4 (T) 11 - T 11 - 1

THETA(I) = the I-th θ -value to be used if THELAM = 1.

LAMBDA(I) = the I-th λ -value to be used if THELAM = 1.

The following relations must always be true:

0 ≤ THETA(1) ≤ THETA(2) ≤ THETA(3)
0 ≤ LAMBDA(1) ≤ LAMBDA(2) ≤ LAMBDA(3).
If THELAM = 0, these values are ignored.

The quantities on card 9 are parameters used by the synthesis program (SYNTH).

9 1-10 IIO IOPT = a computation option.

IOPT = 0 implies the PCT-th
 percentile of the network's approximate

completion time distribution is to be compared to the specified project deadline, PD.

IOPT = 1 implies the mean of the
 network's approximate
 completion time distribu tion is to be compared to
 the specified project
 deadline, PD.

best. A standard value would be TT = PD.

11-20	110	NT = the number of subdivisions in the
		network's synthesized completion time
		distribution; NT < 49.
21-30	F10.5	PCT = the percentile value to be used if
		IOPT = 0; for example, PCT = 90.0.
31-40	F10.5	PD = the specified project deadline.
41-50	F10.5	TT = the initial target time to be used
		by the deterministic project scheduler
		(DPS) in attempting to generate a project
		schedule whose corresponding completion
		time distribution has mean (IOPT = 1)
		or PCT-th percentile (IOPT = 0) equal
		equal to PD. Usually TT < PD works

 $\label{eq:APPENDIX B} \mbox{ Computer Input for the Example in Section 3}$

18	9	1	9	9	
1	1	2	4		
10	15	20	30		
100	70	50	30		
1	-1	5.	.257812	20.	.257812
2	0		17.5	5.	1.
3	-1	10.	25.	40.	-1.
2	1	2	2		
d	14				
60	45				
1	0		11.	3.	1.
3	102	2	2		
10	10				
50	50				
1	2	10.			
4	2	6	2		
50	70				
200	160				
1	0		60.	10.	1.
5	2	5	2		
32	40				
45	40				1
1	0		36.	4.242641	1.
6	2	4	2		
20	32				
64	48				
1	-1	10.	.794285	50.	1.691428
7	2	3	2		
13	19				
30	25				
1	0		16.	2.645751	1.
8	5	6	3		
20	30	35			

60	50	48			
1	0		25.	3.464102	1.
2	0	23.5	32.5	41.5	-1.
9	4	5	2		
5	5				
60	60				
1	2	5.			
10	3	4	2		
18	26				
62	49				
1	-1	10.	13.	40.	20.
11	6	8	2		
10	10				
40	40				
1	2	10.			
12	6	7	2		
6	10				
75	50				
1	0		8.	2.236068	1.
13	5	7	2		
36	44		•		
225	175				
1	0		40.	8.944272	1.
14	4	7	2	0.744212	
30	40		-		
400	300				
1		•			
	0	20.	35.	50.	-1.
15	3	9	2		
60	80				
250	210				
1	0		70.	12.	1.
16	7	8	2		
4	4				
52	52				
1	2	4.			
17	7	9	2		
2	4				
100	90				
1	-1	1.	1.	6.	2.
18	8	9	2		
4	6				
110	80				
1	0		5.	1.414214	1.
1	1	1			
20	9	1	500		
0					
1	49		125.	125.	

Appendix C

Computer Output for the Example in Section 3

STATISTICAL PERT

A PROJECT SCHEDULING PROCEDURE

WRITTEN AT THE

INSTITUTE OF STATISTICS
TEXAS AGM UNIVERSITY
COLLEGE STATION, TEXAS 77843

AUGUST 1978

INQUIRIES AND CONNENTS SHOULD BE ADDRESSED TO: ROBERT L. SIELKEN JR.

THIS IS THE OUTPUT FROM THE MAIN PROGRAM: MAIN

THE PROJECT NETWORK AS IT WAS READ IN:

THE NUMBER OF MODES IS 9.
THE NUMBER OF ACTIVITIES IS 18.
THE SCHROE NODE IS NUMBERED
THE LARGEST NOCE NUMBER IS 9.

.

	0.25781												1.69143									20.0000														2.00000			
	0.25781. BETA=		000000										0.79428. BETA.					41.50000				13.00000. BETA.								20.00000						1.0000C. BETA=			
	20.00000. ALPHA:	000000	25.00000. MAX=		3.00000				0000001		4.24264		50.00000. ALPHA=		2.64575		3.46410	32.50009. MAX=				40-00000. ALPHA=				2.23607		8.54427		35.00000. MAX=		12.00000				6.00000. ALPHA=		1.41421	
	5.00000. HAX=	17-50000. ST.DEV=	10.00000. MODE=		11.00000. ST.DEV=		10.00000		60.00000. ST.DEV.		36.00000. ST.DEV*		10.00000. MAXE		16.00000. ST.DEV=		25.00000. ST.DEV=	23.50000. MEAN=		5.00000		10.00000 MAX=		10.00000		8.00000. ST.DEV=		40.00000. ST.DEV=		20.00000 MEANS		70.00000. ST.DEV=		**00000		1.00000. MAX=		5.00000. ST.DEV=	
PARAMETERS		MEANS	MIN.		MEANS		TIME		MEAN 6		MEANS 3				MEANE		MEAN# 2	MINE 2		TIME		HIN.		TIME= 1		MEAN		MEANS		MIN= 2		MEAN# 7		TIMER		MIN.		MEANE	
DISTRIBUTION	100 BETA	NORMAL	BETA		NORMAL		FIXEDICONSTANT		NORMAL		NOFWAL		BETA		NORMAL		NORFAL	NORMAL		FIXEDICONSTANT		BETA		FIXED(CONSTANT)		LORMAL		NORMAL		CORPAL		NGFFAL		FIXEDICONSTANT		BETA		NORMAL	
CCST	100	20	20	30	00	•	20	20	200	160	5.	•	•	8 4	30	52	00	20	8.	00	00	62	64	0	•	75	20	555	175	200	300	250	210	25	25	100	05	110	80
11 ME																																							•
7	1 2														3 -																			-	8 2	-	9 2	- 6	8 5
R 16 TE	-	-	-	-	-	-	-	-	~	~	~	~	~	~	~	~	•	•	5			•	-	•	•		•	•	•		•	~	•				-	•	
ACTIVITY CRIG TERM	-			-		~	•	•			•	•	•	•					•	•	•	10	2	=	=	1.2	12	13	-	:	•	**	15	16	91	11	11	16	97

THE INFORMATION INPLY TO THE DETERMINISTIC PROJECT SCHEDULING PROGRAM WILL NOT BE PRINTED. THE INTERMEDIATE CUTPUT GENERATED BY THE DETERMINISTIC PROJECT SCHEDULING PROGRAM WILL NOT BE PRINTED.

THE SIMPLIFICATION PREGRAM SILL PRINT ONLY THE MINIMUM OUTPUT.

DURING THE SUBMETWORK ANALYSIS PROCRAM.

THE LARCEST WHERE CF SUBDIVISIONS IN THE ESTIMATED DURATION DISTRIBUTION FOR EACH SUBMETWORK WILL BE 20:

THE LARCEST WHERE CF ACTIVITIES IN A SUBMETWORK FOR WHICH THE SUBMETWORK'S

EXACT DISCHERE CHARTICN DISTRIBUTION WILL BE ENUMERATED IS 9:

THE MAXIMUM NUMBER OF ACTIVITY DURATION COMPIGERED INTHE DETERMINATION OF THE BOUNDS:

THE UNICONCLUSTER PROCEDURE MILL BE USED IN THE FORMATION OF THE BOUNDS:

THE FELLGMING (THETA.LAMBDA) PAINS WILL BE USED IN THE FORMATION CF THE CLUSTERS:

(1.000000. 1.000000).(2.000000).(3.00000).

THE

THE NUMBER OF SUBBIVISIONS IN THE NETWORK'S SYNTHESIZED COMPLETION TIME DISTRIBUTION WILL BE 49. THE MEAN OF THE NETROCKY'S APPROXIPATE COMPLETION TIME DISTRIBUTION WILL BE COMPARED TO THE SPECIFIED PROJECT DEADLINE OF 125,00000.

THE TARGET TIME FOR THE FIRST ITERATION IS 125.00000.

THIS IS THE OUTPUT FROM SUBPROGRAM LOOP. THE PROGRAM SEARCHES THE GIVEN PROJECT NETWORK FOR LOOPS (CYCLES). A VALID PROJECT NETWORK SHOULD CONTAIN NO LOOPS.

NGCE 1 15 NOT PART CF A LCCP
NGCE 2 15 NOT PART CF A LCCP
NGCE 4 15 NCT PART CF A LCCF
NGCE 5 15 NGT PART CF A LCCF
NGCE 6 15 NGT PART CF A LCCF
NGCE 6 15 NGT PART CF A LCCF
NGCE 6 15 NGT PART GF A LCCF
NGCE 6 15 NGT PART GF A LCCP
NGCE 6 15 NGT PART GF A LCCP
NGCE 6 15 NGT PART GF A LCCP
NGCE 7 15 NGT PART GF A LCCP

THIS IS THE GUTPUT FACH THE DETERMINISTIC PROJECT SCHEDULER: DPS	ISTIC PROJECT SCHEDULER: DPS	
THE ENTIRE PROJECT COST CLAVE WILL BE	ILL BE DETERMINED.	
LAPBEA = FRCJECT COMPLETION TIME		
THE STARTING VALUE OF LAMBDA IS	135.	
THE CCRRESFENDING TGTAL PROJECT COST IS	IS 0.15520E 04.	
CELTA (REFRESENTED BY "D") FANGES FROM 0 TO 1. LABBOA RANGES FRCM 135 TC 134. THE BINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE	134. 134. RROJECT DEADLINE = 135-D:	
PACJECT CEMPLETICN TIME = 135-0.	· c	
ACTIVITY 8: 1 NEB VALUE: XACT(1)	ACTIVITY COST	
00		
***	0.450006	
. 2		
•		
32		
35-0	0.48000E 02 + (0.40000E 00*D)	
5 46	0.60000E 02	
27.	0.5000E 02	
•		
· TOTAL STATE STAT		
91	0.520006 02	
THE CLARENT VALUE OF THE PREJECT COST IS	15 0.15520E 04 + (0.40015E 00#D).	
BY "C") RANGE	S FROM 0 TO 6.	
THE MINIMUM COST PROJECT SCHEDULE FOR	F FOL DEADLINE = 134-0:	
	-	
PACJECT CCHFLETION TIME = 134-D	·	
ACTIVITY 8: 1 NES VALUE: XACT(1)		
7	0.450006 02	
• •	0.16000E 03	

0.63333E 0000)	0.63333E 000D).	0-16250E 01-00)	0.16250E 01*01.
0.2000E 02 0.50000E 02 0.60000E 02 0.50000E 02 0.50000E 02 0.175000E 03 0.51000E 03 0.52000E 03 0.52000E 03	0.15524E 04 + (ACTIVITY COST 0.30000E 02 0.50000E 02 0.10000E 02 0.40000E 02	10 TO 10. 114. PROJECT DEADLINE = ACTIVITY COST 0.30000E 02 0.45000E 02 0.45000E 02 0.45000E 02 0.45000E 02
7 - 4 4 5 5 4 4 6 4 4 6	THE PRGJECT CGST IS T =0=1 RANGES FRCH 0 TO 124 TG 126 TG 124 LECT SCHEDULE FOR PROJECT INE = 120-D.	NEW VALUE: XACT(!!) 30 40 50 50 50 60 60 60 60 60 60 60 60 60 60 60 60 60	E PRCJECT CGST 124 TG 124 TG 1 SCHEDULE FOR 1 SCHEDULE FOR 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
*********	THE CLARENT VALUE OF THE PROJECT COST IS DELTA (REFRESENTED BY "O") RANGES FROM O LAWEDA RANGES FROW 128 TG THE MINIMUM COST PROJECT SCHEDULE FOR PROPELECT COMPLETION TIME = 128-D.	ACTIVITY 6: 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	THE CLARENT VALUE OF THE PROJECT COST IS DELTA (REFRSENTED BY "D") RANGES FROM O LAMBDA RANGES FROM 124 TO INTE BININUM COST PROJECT SCHEDULE FOR PA PROJECT COMFLETION TIME = 124-D. ACTIVITY A: I NEW VALUE: MACT(I) 1 2 3 4 5 5 6 6 6 70 6 6

									0.20000E 01.D).			:0-+11						0.42500F 00#D)				10000	0-10-20E 01-01							0.22500E 01.00).		113-0:				10-10 2000-0				
0.30000E 02 0.48400E 02					0 3000E 03			0.80000E 02	0-15639E 04 + (-1		ECT DEADLINE .		ACTIVITY COST			0.5000dE 02		05			0.600005 02	200		0.17500E 03		0.21666E 03		0.8000CE 02	0.15839E 04 + (0	•	CT DEACLINE =		***************************************						
24 *		=======================================	2 :	01	: *	3	THE RESERVE TO SECURITION AND ADDRESS.	• 0	THE CURRENT VALUE OF THE PROJECT COST IS	SY "D" J RANGES FROM 0 TO		JECT SCHEDULE FOR PROJECT DEADLINE	IIME = 114-0.	NEB VALUE: XACT(I)	20	•	0 :	8-9	35	13	**	5	0-22	01	:	0 ;	0			THE CLARENT VALUE OF THE PRCJECT COST IS	CELTA (REFRESENTED BY "C") MANGES FROM 0 TO LABBOA RANCES FROM 113 TO 108.	LE FCR	1ME = 113-0.		NEW VALUE: AACTOO	7-02	 202	35	32	13
~ • •	• •	.:	::	2:	2:	::		- =	THE CURRENT VALUE C	DELTA (REFRESENTED BY	LAMECA RANGES FROM	THE PINIBLE COST PROJECT	PRCJECT CCMFLETICN TIME	ACTIVITY 6: 1 NE		2	7 .	• •	•		•	• •	2:	12		::	- :	1.1	•	THE CLARENT VALUE OF	CELTA (REFRESENTED ELABOR PANCES FROM	THE MINIMUM CCST PRO	PRCJECT CCPPLETICN TIME				•	•	•	,

	0.40000E 014D1.	0.62500E 00*D)	0.42500E 010D). 106-D: 0.62500E 000D)
	• •	ACTIVITY COST 0.70000E 02 0.4000E 02 0.4000E 02 0.4000E 02 0.4000E 02 0.4000E 02 0.4000E 02 0.5000E 02 0.5000E 03 0.21000E 03 0.21000E 03 0.2000E 03 0.2000E 03 0.2000E 03 0.2000E 03 0.2000E 03 0.2000E 03	00
	CURRENT VALUE OF THE PROJECT COST IS 0.15861E A (REFRESENTED BY "D") RANGES FROM 0 TO 2. GA RANGES FROM 108 TO 106. PINIMUM CCST PRCJECT SCPEDULE FOR PROJECT DEADLINE JECT CCWFLETICN TIME = 106-D.	AES VALUE: XACT(!!) 15 10 10 36 36 13 10 10 10 10 60-D	THE CLARENT VALUE OF THE PROJECT COST IS 0.16061E CELTA (REFRESENTED BY "D") RANGES FROM 0 TO 1. LAMBGA RANGES FROM 106 TO 105. LAMBGA RANGES FROM 106 TO 1. LAMBGA RANGES FROM 106 TO 1. LAMBGA RANGES FROM 0 TO 1. ACTIVITY 8: I NEW VALUE: XACT(1) ACTIVITY 0.700001 A ACTIVITY 8: I NEW VALUE: XACT(1) 0.4500001 A A ACTIVITY 0.1 10 0.4500001 A A ACTIVITY 0.1 10 0.4500001 A A ACTIVITY 0.1 10 0.4800001 A A ACTIVITY 0.1 10 0.4800001 A A ACTIVITY 0.1 10 0.4800001 A A ACTIVITY 0.1 10 0.4840001
** 2 : 2 : 2 : 2 : 2	THE CURRENT VALUE OF THE PROJECT DELTA (REFRESENTED BY WOW) RANGES LAWGLA RANGES FROM 108 TO THE WINIHUM COST PROJECT SCHEDULE PROJECT COMPLETION TIME =		THE CLARENT VALUE OF THE PROJECT COST CELTA (REFESENTED BY "0") RANGES FROM LAMBCA RANGES FROM THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT COMFLETION TIME = 106-1 ACTIVITY 8: I NEW VALUE: XACT(1) 1 ACTIVITY 8: I NEW VALUE: X

01•0)	0 10	• 60		(0010					,					.10010					0010					
0.16250E	0.20000E 01.D)	0.55833E 01.00).		0.60000E 01.D)										0.60000E		104-0:			0.250006					
	0.50000E 02 0.17500E 03 0.21400E 03 0.52000E 02 0.90000E 02	0.16146E 04 + (ACTIVITY COST 0-70000E 02 + (0.42500E 02 0.49333E 02	0.48400E 02		0.62000E 02			0.21600E 03		0.90000E 02 0.80000E 02	0.16202E 04 + (:	ECT DEADLINE =		1SC	0.45000E 02 + (0.16000E 03			
n 1 2 2	0 4 9 6 4 4 0	VALUE OF THE PROJECT COST IS 0.10146E SENTED BY "O") RANGES FROM 0 TO 1. S FROM 105 TO 104. CCST PROJECT SCHEDULE FOR PROJECT DEADLINE	TIME = 105-D.	NEB VALUE: XACT(1)	 3.6	£ 6	'n	0 0		:	0.2	•	••	THE CLRRENT VALUE OF THE PROJECT COST 15	ES FROM	DIECT SCHEDULE FOR PROJECT DEADLINE	11#6 = 104-0.	MEN VALUE: XACT(!)	0-4-	01	0 4	3 7	2	* "
*2 = 2	224944	THE CLARENT VALUE OF THE CELTA (REPRESENTED BY "O" LAWECA RANGES FROM THE PINIBLE CST PROJECT	FRCJECT CCWFLETICN TIME	ACTIVITY 6: 1 N	 	~ •	•	2 =	2		: ::	2	::	THE CLARENT VALUE CF	CELTA (REFRESENTED E	THE WINIMUM COST PROJECT	PREJECT CEMPLETION TIME	ACTIVITY S: 1 M	- 7	•	•••	. •		• •

	:	ç c	6 6	•	0 0
	•	: 8	0 0	50	9 10
	0.85000E 01.0D!	0.20000E 01#D)	0.62500E 01#D)	0.10650E 02eD).	0.20000E 01*D)
22222222	•	+ + +	+ + + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	:	+ + + + + + + + + + + + + + + + + + + +
	5 #	005000000000000000000000000000000000000		8 .	000000000000000000000000000000000000000
0.62000E 0.17500E 0.31600E 0.5200E 0.5200E	0.16262E	ACTIVITY COST 0.10000E 03 0.55000E 02 0.16000E 03 0.42500E 02 0.42500E 02 0.42500E 02	0.5000E 0.17500E 0.30000E 0.30000E 0.52000E	0.16602E	ACTIVITY CDST 0.10000E 03 0.55000E 02 0.1660E 02 0.42500E 02 0.45333E 02 0.30000E 02 0.50000E 02
800779	6 8	100000000000000000000000000000000000000	80551011005	8 8	1 × 1 × 1 × 1 × 1 × 1 × 1 × 1 × 1 × 1 ×
		500000000		96 ÷ 9.	5000000000
50040 <u></u>	FECJECT COST 1S 0.16262E 1) HANGES FROM 0 TO 4. 100 TO 96. SCHEDULE FOR PROJECT DEADLINE	9 9	, q-0-4-4-0	PRCJECT CGST IS 1) RAAGES FROM 0 TO 96 TC 92. SCHEDULE FCR PROJECT	XACT(1) 10 10 66-0 31 31 30-0
	FECJEC P RANG 100 TO SCHEDU			RCJEC RANG 96 TG Cheou	
		3			3
	BY PE	ALUE:		BY "C ROJECT	NE VALLE:
2-22-22-2	THE CLARENT VALUE OF THE FECJECT COST IS DELTA (REFRESENTED BY "G") MANGES FROM 0 TO LAMBGA FANGES FROM 100 TO THE MINIBUM CCST PROJECT SCHEDULE FOR PROJECT PROJECT (CYPLETION TIME = 100-0.	ACTIVITY	· 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2	THE CLRRENT VALUE OF THE PRCJECT CGST IS DELTA (REFRESENTED BY "D") RANGES FROM 0 TO LAWELA RANGES FROM 96 TC 92. THE MINIMUM CGST PRGJECT SCHEOLLE FGR PROJECT PRCJECT CCPFLETICN TIME = 96-D.	ACTIVITY 6: 1

	0100		01.00				05.001.																			01.00			05.0)
	0.62500E 01*D)		0.20000E 01 0)				0.11250E 0240).		95-0:																	0.20000£ 01*01			0.15000E 02*D)
	-		-																							-			-
95	. 60	03	03 +	20	05	05	•				_	63	05	05	03	05	05	02	05	05	05	05	02	03	03	• 60	05	05	95 +
							:				503																		
000	0.2000E	0.30000E	0.22400E	0.52000E	300006 °	0.80000E	0.17028E 04 + (LAWBDA RANGES FROM 92 TO 90. The Winimum COST PROJECT SCHEDULE FOR PROJECT DEADLINE =		ACTIVITY COST	0.10000E	0.55000E	0.50000E	0.17600E	0.42500E	0.49333E	0.30000E	0.54000E	0.60000E	0.620005	0.4000E	0.50000E	0.22500E	0.30000E	0.23200E	0.5200CE	300005.0	900008-0
0.50000E	500	30	.22	520	800	. 80	170		10		7	10	55	200	17	. 45	•	300	54	09	.62		200	55	30	.23	52	06	900
•	ö	ó	ò	ö	•	·			DEA		13	ò	•	ċ	ò	ċ	ċ	•	•	·	ė	ċ	ö	•	ċ	ċ	•	ė	•
									5		*																		
								2	90. ROJE																				
							15		• 4	·																			
							DST	68	8	92-0-	=																		
	o		o				5	S	w		1															o			0-9
2	0-04	0	73-0	•	•	•	EC	34	92 TO Crebut		NEW VALUE: XACT(1)	9	0	2	9	36	7	2	50	0	9	9	2	26		0-60	•	•	٥
							2	è	25																				
								:	5		7																		
							ž	:) FC	¥.	-																		
							5	6	- 6	-	NE																		
							3	1	2 -	9																			
							THE CURRENT VALUE OF THE PROJECT COST 1S	DELTA (REFESENTED BY "C") FANCES FROM 0 TO	LAWBOA RANGES FRCH The Winimum COST P	PROJECT CCMPLETION TIME .	ACTIVITY 6: 1																		
							7	FEE	9 3	3	,																		
12	2	:	15	-	-	2	an E	BE	1 5	-	=	-	N	~	•	6	•	-	•	•	2	=	12	7	•	5	=	-	-
							3		5	E	=																		

THE SINK BAS REACHEC BITH INFINITE CAPACITY IMPLYING AN INFEASIBLE SOLUTION TO THE PRIPAL PROBLEM
1F LAMBDA OROPS BELOW ITS CURRENT VALUE.

THIS IS THE GUTPUT FACH THE DETERMINISTIC SCHEDULE RESOLUTION PROGRAM: DSR

.5:	
FOLLO	
IS AS	
125.00000 IS AS FOLLOWS:	
90	
TIME OF	
TARGET T	
FCR	
THE MINIMUM COST PROJECT SCHEDULE FOR A 1	
PROJECT	
COST	
PINIBLE	
17.	

MEAN	DISTRIBUTION	PARAMETERS	ERS						
-	BETA	H N W	12.00000, MAX=	MAX=	48.00000. ALPHA=	AL PHA=	3.00000. BETA=	BETA	3.00000
-	COEMAL	ME ANE	14.00000. ST.DEV=	ST.DEV=	3.81818				
-	FIXECICCUSTANT	TIME	10.00000						
-	CHMAL	MEANE	70.00000. ST.DEV=	ST.DEV=	11.66666				
-	UENAL	MEANE	40.00000. ST.DEV=	ST.DEV=	4.71404				
-	BETA	ZZZ	12.30770.	MAX=	61.53848. ALPHA=	ALPHA=	0.79428. BETA:	BETAS	1.69143
-	CEMAL	ME AN=	13.00000. ST.DEV=	ST.DEV=	2.14967				
-	CEMAL	MEANE	34.00000. ST.DEV=	ST.DEV=	3.13846				
	1 XEDICONSTANT)	TIME	2.00000						
-	BETA	II Z	10.45455.	MAXE	41.81821. ALPHA=	ALPHA	12.99998, BETA=	BETAF	2000002
	TXECICCUSTANT	TIME	10.00000						
	HORMAL	REANE	10.000000 ST.DEV=	ST.DEV=	2.79508				
	OFFIR	MEANE	44.000000. ST.DEV=	ST . DE V=	9.83869				
-	LORMAL	MEANE	40.00000. ST.DEV=	ST.DEV=	5.71428				
	MOKWAL	MEANE	80.00000. ST.DEV=	ST.DEV=	13.71428				
	TXEC(CONSTANT)	TIME	4.00000						
-	BETA	=NIN	1.33333. MAX=	MAX=	8.00000. ALPHA=	AL PHA=	1.00000. BETA=	BETAR	2.00000
	CHARL	MEANE	6.00000. ST.DEV=	ST.DEV=	1.69705				

THIS IS THE CUTPUT FROM THE SIMPLIFICATION PROGRAM: SIMP

THE FCLLCWING ACTIVITIES CANNOT BE FURTHER COMBINED:

CTIVITY	CAIGIN	TERMINAL	ACTIVITY NUMBER ASSIGNED BY DECOMP	ASSIGNED	9	-
	٠.	. •	. 7			
•	~	•	•			
•	~		•			
	*	•	ø			
•	•	•	•			
•		•				
•	•		•			
=	•	•	•			
	•		10			
2	•					
:	•	1	12			
15	•	•	27			
*	-	•	:			
		•	1.5			
	•	•				

Market 1				
CRIGIN TERMIMAL MEAN CRIGIN MEAN CRIGIN TERMIMAL MEAN CRIGIN TE	ACTIVITY	CRIGIN	TERMINAL 2	STANDARD DEVIATION
CRIGIN CRIGIN			1 103	
CRIGIN TERMINAL MEAN			1	
CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN FILT T T T T T T T T T T T T T T T T T T				
CRIGIN TERMIMAL MEAN			1	
CRIGIN TERMINAL NEAN CRIGIN C. 25 29.1490 0.455 33.0000 0.455 33.0000 0.455 37.2000 0.455 44.4000 0.456 44.4000 0.456 44.4000 0.456 44.4000 0.456 44.4000 0.456 44.4000 0.456 65.546 0.456 65.546 0.456 65.546 0.456 65.546 0.456 65.546 0.456 65.546 0.456 65.546 0.456 65.546 0.456 65.546 0.456 65.546 0.456 65.546			1	
CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN FIT 1 T 000 0.05 57.000 0.05 57.000 0.05 57.000 0.05 57.000 0.05 65.5046 0.05 67.001 0.05 67.001 0.05 67.001 0.05 67.001			1	
CRIGIN TERMINAL MEAN 2			!!	
CRIGIN TERMINAL 44.4000 1.00 44.4000 2			11	
CRIGIN TERMINAL 40.000 FITT TO 35.000 0.05 57.000 0.25 57.000 0.45 68.540 0.45 68.540 0.45 82.011 0.45 82.011			1	
CRIGIN TERMINAL MEAN FITT T 0.0 35.0000 0.05 57.0001 0.25 62.5046 0.45 68.5046 0.45 71.4660 0.65 71.4660 0.65 71.4660 0.65 71.4660			1	
FILT TERMINAL MEAN CRIGIN TERMINAL MEAN CRIGIN CO.O. T. 135,00000 CO.O. C. 135,00000 CO.O				
FILT TERMINAL MEAN TO.0 C.0 C.0 C.0 C.0 C.0 C.0 C.0 C.0 C.0 C				
1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	ACTIVITY	CRIGIN	TERMINAL	STANDARD DEVIATION
11111111111		~		10°04
			!	
			! !	
1111111			!	
11111			!	
1111			!	
1111			!	
111			-	
!!			!	
1			!	
			0.55 89.1908	

STANDARD DEVIATION													STANDARD CEVIATION	10.03													STANDARD DEVIATION	2.02													
TERMINAL MEAN	26.0670	1 !	1	0.25 - 36.8204	;	:	0.55 - 40.5924	:	1	1	1	1.00 54.1421	TERMINAL MEAN	4 31.9	 0.0 - 12.3077	1	1	1	0.35 26.8016	1	1	1	1	1	0.95 50.1376	1.00 61.5385	TERMINAL	3 13.0	1 100	0.0 6.5510	-	-	1	1	!	!	0.65 13.8283	!	0.85 15.2280	1	1.00 19.4490
CRIGIN													CRIGIN	~													CRIGIN	~													
ACTIVITY													ACTIVITY	•													ACTIVITY														

F(I) T 24.5846 0.05 24.5846 0.05 28.6375 0.15 30.7471 0.25 33.7907 0.45 33.7907 0.45 33.1944 0.65 35.2043 0.65 35.2043 0.65 35.2043 0.65 35.2043 0.65 35.2043 0.65 36.1109 0.65 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 10.4546 0.05 10.4546 0.05 10.4546 0.05 10.4547 0.05 10.4547 0.05 10.4547 0.05 10.4547 0.05 10.4546 0.05 10.4547 0.05	ACTIVITY	ORIGIN	TERMINAL	NAL MEAN	STANDARD DEVIATION
CRIGIN CRIGIN		s	•		2.94
CRIGIN CRIGIN			F(T)	•	
CRIGIN TERMINAL MEAN CRICIN TERMINAL MEAN			0.0	24.5846	
CRIGIN TERMINAL MEAN CAS 31.8431 C.25 5.0000 C.25			0.05	28-4375	
CRIGIN TERMINAL MEAN			0.15	30.7471	
CRIGIN CRIGIN			0.25	31.8831	
CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN CRICIN TERMINAL MEAN			0.35	32.7907	
CRIGIN TERMINAL MEAN			0.45	33.6056	
CRIGIN TERWINAL MEAN CRIGIN TERMINAL MEAN			0.55	34.3944	
CRIGIN TERMINAL MEAN FILT) FILT) FILT) FILT) FOOD COSS COSS			0.65	35.2093	
CRIGIN TERMINAL MEAN F(17) T 0.00 0.05 39.1025 1.00 43.4154 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 1.00 5.0000 0.05 5.0000 1.00 5.0000 0.05 10.4546 0.05 10.4546 0.05 10.4546 0.05 10.4546 0.05 10.4546 0.05 10.4546 0.05 10.4546 0.05 10.4546 0.05 10.4546 0.05 20.3169 0.05 10.4542 0.05 21.9550 0.05 22.6922 0.05 22.6920 0.05 22.6920 0.05 22.6920 0.05 22.6920 0.05 22.6920 0.05 22.2590 0.05 22.2590 0.05 22.2590 0.05 27.3216			0.75	36-1109	
CRIGIN TERMINAL MEAN FILL O.0 - 5.0000 O.05 - 23.000 O.05 - 23				37.2529	
CRIGIN TERMINAL MEAN F(T) 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 1.00 5.0000 1.00 5.0000 1.00 5.0000 1.00 5.0000 1.00 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 5.0000 0.05 21.9427 0.05 21.9427 0.05 21.9427 0.05 21.9427 0.05 21.9427 0.05 21.9427 0.05 21.9427 0.05 21.9427 0.05 22.0100 0.05 22.0100 0.05 22.0100 0.05 22.0100			0.95	39.1625	
CRIGIN TERMINAL MEAN FILT) FILT) O-05				43.4154	
CRIGIN TERMINAL MEAN FIT) 0.0 - 5.0000 0.05 - 5.0000 0.05 - 5.0000 0.05 - 5.0000 0.05 - 5.0000 0.05 - 5.0000 0.05 - 5.0000 0.05 - 5.0000 1.00 - 5.0000 1.00 - 5.0000 1.00 - 10.4546 0.05 - 20.3169 0.05 - 21.2104 0.05 - 21.2000					
FIT) T 5.0000 0.05 5.0000 0.15 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 21.2104 0.45 21.2104 0.45 21.2104 0.45 21.204 0.45 21.204 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206 0.45 21.206	TIVITY	CRIGIN	TERMIN		STANDARD DEVLATION
CRIGIN TERMINAL MEAN CRIGIN TERMINAL TERM	•	•	•		00.0
CRIGIN TERMINAL MEAN 5.0000 0.15 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 1.00 5.0000 1.00 5.0000 0.45 5.0000 0.45 5.0000 0.45 21.2104 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250 0.45 21.250			613	-	
CRIGIN CRIGIN			0.0	2.0000	
CRICIN CRICIN			0.05	2.0000	
CALCIN TERMINAL MEAN CALCIN TERMINAL MEAN FITTH CALCIN TERMINAL S.0000 0.05 5.0000 1.00 5.0000 FITTH TO 0.0 10.4546 0.05 10.4546 0.05 10.4546 0.05 10.4546 0.05 20.3169 0.05 21.9427 0.05 22.6122			0-15	2.0000	,
CFIGIN TERMINAL MEAN FITT TERMINAL MEAN FITT TERMINAL MEAN 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 5.0000 0.45 2.0000			0.25	00000-5	
GRIGIN TERMINAL MEAN GRIGIN TERMINAL MEAN FITT TERMINAL MEAN 1.00 - 10.4546 0.00 - 10.4546			0.35	2.0000	
CRICIN TERMINAL MEAN S. CRICIN TERMINAL MEAN FITT TERMINAL MEAN S. CRICIN TERMINAL MEAN S. C			0.45	2.0000	
CRIGIN TERMINAL MEAN FITTH FITTH O.0 - 10.454 0.45 - 5.0000 1.00 - 5.0000 FITTH O.0 - 10.454 0.45 - 20.3169 0.45 - 21.9427 0.45 - 22.452 0.45 - 22.452 0.45 - 22.452 0.45 - 22.452 0.45 - 22.452 0.45 - 22.452 0.45 - 22.452 0.45 - 22.452 0.45 - 22.452 0.45 - 23.4950 0.45 - 27.3216			0.55	2.0000	
GRIGIN TERMINAL MEAN STATE OF			-	2.0000	
GRIGIN TERWINAL MEAN 3 FITT TERWINAL MEAN 3 0.0 10.4546 0.05 10.4546 0.15 21.2104 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690 0.45 21.2690			0.75 -	2.0000	
CALCIN TERMINAL MEAN 3 F(T) T 23.00 6.05 10.4546 0.05 10.4546 0.25 20.3169 0.25 21.9427 0.45 22.6422 0.55 23.4550 0.55 23.4550 0.55 23.4550 0.65 23.4550 0.65 23.4550 0.65 23.4550			0.65	2.0000	
GRIGIN TERMINAL MEAN 3 FITT TO 0.0 0.05 10.4546 0.45 10.4546 0.45 21.2104 0.45 21.2104 0.45 23.2590 0.45 23.2590 0.45 23.2590 0.45 23.2590 0.45 23.2590 0.45 23.2590 0.45 23.2590 0.45 23.2590			0.65	2.0000	
FITTH T 23.0 FI			1.00	2.0000	
F(T) TERMINAL MEAN 23.0 F(T) T 0.0 10.4546 0.15 10.4546 0.15 21.2104 0.25 21.2104 0.35 21.2692 0.55 21.2692 0.55 21.2692 0.65 21.9595 0.65 21.9595 0.95 21.9595 0.95 21.9595 0.95 21.9595 0.95 21.9595 0.95 21.9595 0.95 27.3216					
F(T) T 0.00 10.4546 0.05 18.4855 0.15 20.3169 0.25 21.427 0.45 21.427 0.45 21.427 0.45 21.45 0.05 21.45 0.05 21.45 0.05 21.45 0.05 21.45 0.05 21.45 0.05 0.45 21.45 0.05 0.05 21.45 0.05 0.05 21.45 0.05 0.05 21.45 0.05 0.05 0.05 21.45 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0	TIVITY	CRIGIN	TERMIN		STANDARD CEVIATION
11111111111	01	•	•		2.42
11111111111			FUT	•	
1111111111			1 0:0	10.4546	
111111111			0.05	18.8855	
11111111			0-15	20.3169	
1111111			-	21.2104	
111111			0.35	21.9427	
11111			54.0	22.6122	
1111			- 5	23.2690	
111				23.9550	
11			0.75	24.7250	
1				25.0542	
				27.3216	

STANDARD DEVIATION	00.00														MOTATIVE CRACKET	2.62														STANDARD DEVIATION	9.23													
	10.0		10.000	10-0000	10.0000	10.0000	10.000	10.000	10.0000	10.0000	10.000	10.000	10.0000	10.000	AL MEAN			1.6147	5.4023	7.1030	8-1147	8.9230	9.6488	10.3512	11.0770	11.8653	12.8970	14.5977	18.3653	AL MEAN			14.4839	27,8160	33.8027	37.3639	40.2050	42.7637	45.2363	47.7910	50.0361	54-1973	60-1840	73.5161
TERMINAL		5.13	0.0	0.05	0.15	0.25	0.35 -	0.45	0.55	0.65	0.75	0.85	0.95	1-00-1	TERMINAL	-	500	1 0.0	0.05	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95	1.00	TERMINAL		F(T)	0.0	0.05	- 51.0	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.55	1.00
OR 161N															CRIGIN	•														OR 161N	'n													,
ACTIVITY	=														ACTIVITY	71														ACTIVITY	2													

110h															NOI															NOI													
STANDARD DEVIATION	5.36														STANDARD DEVIATION	12.86														STANDARD DEVIATION	00.0												
IAL MEAN	• • •		22.8572	30.6004	34.0774	36-1458	37.7982	39.2819	40.7181	42.2018	43.8542	45.9226	49.3996	57-1428		80.0	-	38.8572	57.4410	65.7858	10.7499	74.7156	78.2767	81.7233	85.2844	89.2501	94.2142	121-1428			•••	-	4.0000	4.0000	4.0000	** 0000	***	**0000	****	00000	00000	00000	00000
TERMINAL	•	100	0.0		0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	56.0	1.00	TERMINAL	•	F(T)	0.0	0.05	0-15	0.25	0.35	0.45	0.55	0.65	0.75	1 69.0			TERMINAL	•	500	0.0	0.05	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	-
ORIGIN															CRIGIN	•														ORIGIN													
ACTIVITY	:														ACTIVITY	2														ACTIVITY	9												

STANDARD CEVIATION														STANDARD CEVIATION	1.59													
TERMINAL MEAN		- 1.3333	!	!	!	1	1	6.0679	5 4.5038	5 4.9579	!	1	00000-0	TERMINAL MEAN	0.0		0.9088	1	5 - 4.2411	\$ 4.8554	5.3461	5 - 5.7667	5 - 6.2133	5 - 6.6539	5 - 7.1446	1	1	11.0512
CRIGIN	1111	0.0	0.05	91.0	0.25	0.35	0.45	0.55	0.65	0.75	0.85	9.95	1.00	CRIGIN		FCT.	0.0	90.0	61.0	0.25	0.35	0.45	0.55	9.65	0.75	0.65	95.0	1.00
40114117														ACTIVITY	•													

	THIS IS THE GUIDUT FNOW THE DECCMPOSITION PROGNAM: DECOMP		INPLT STAGE	THE SIMPLIFIED NETLCHK PAS 16 ARCS	THE SCIENCE IS NOCE NUMBER	THE SINK IS ACOE NUMBER 9	THE LARGEST NCDE IS NOCE NUMBER 9	THE SIMPLIFIED NETWORK AS REAC IN 1S:	ANC NUMBER CRIGIN ACCE TERRIMAL MODE 2
--	---	--	-------------	------------------------------------	----------------------------	---------------------------	-----------------------------------	---------------------------------------	---

STAGE 2 BREAKLP

SUBNETRORK 2 IS A NON-DECOMPOSABLE NETWORK IT IS COMPOSED OF:

SCURCE NOCE .

SINK NGOE

THE SUBNETHERK AS IT IS READ IN:

ARC STARCI TTARCI

THE FOLLCHING REPRESENTATION OF THE SUBNETHORK ABOVE WILL BE USED BY THE SUBNETHORK ANALYSIS PROGRAM;

ACTIVITY

SUBNETBUCK 3 IS A NON-DECOMPOSABLE NETBURK IT IS COMPUSED OF:

SCURCE NCCE = 2

SINK NCDE . 9

THE SUBNETBUCKE AS IT IS READ IN:

TEARCE	•	6	•	•	•	~	•		•	~	•	•	۰	•	•
SLARCI	~	~	~	~	•	•	•	•	•	•	•	•	•		•
ARC	~	~	•	•	•	•	•	=	-	12	•	2	•	•	•

THE FCLLCBING REPRESENTATION OF THE SUBNETHORK ABOVE WILL BE USED BY THE SUBNETHORK ANALYSIS PROGRAM:

HEAD	n	•	•	8		•	o	•	•	•	n	2	•		•	
TAIL	-	-	-	-	•	•	•	•	•	•	~	~		•	•	
ACTIVITY	-	~	•	•	•	•			•	10	=	12	-	:	15	

THE NUMBER OF NEW-DECOMPOSAR E SUBMETHORIS 15 2

THIS IS THE GUTPUT FROM THE SLUNETBORK ANALYSIS PROGRAM: SUBNET

EACH SUBNETBORK'S APPROXIMATE DURATION DISTRIBUTION IS DETERMINED.

MEAN STANDARD DEVIATION PROB. LOWER PT. 30.8008 6.3962 0.47516 TERMINAL LOWER POINT UPPER POINT 2 24.0785 36.8868 ACTIVITY CRIGIA

INITIAL INPLT

SUBNETBORK 2

SUBNETBORK
FOR
TICA
DISTRIBU
0157
NOT
CLRATION
T.

1													
	101		10-	00	00	00	00	00	00	00	00	00	10
	DISTRIBUTION	0.0	0-200000-0	0.150000	0.250000	0.350000	0.450000	0.550000	0.0000900	0.750000	0.850000	0.950000	0.100000
		0.2	0.2	05	0.5	0.2	0.5	20	20	05	62	20	20
	7116	0.156000	0.200880	0.233460	0.255850	0.274360	6.251490	0.336000	0.336000	0.372000	0.37266	0.0000	30000000

ORIGIN	TERMINAL	S9-0567	UPPER POINT	MEAN	STANDARD DEVIATION	PROB. LOWER PT.
-	•	35.57.82		00000	10.4433	0.50000
		-	0170	00000	4.4218	0.50000
	,	23.0185	43.2217	31.9497	10-0134	
-	2	10.9836	15.0164		2000	0.33743
•		00000	0000	00000	******	0.0000
			00000	10.0000	0.0	1.00000
,	•	1.3782	12.6218	10.0000	2.6218	0.5000
•	•	31.0561	36.9439	34.0000	2.0410	00000
	•	34.7713	53.2287	00000	0. 2267	000000
•		5-0000			107711	000000
			00000	000000	0.0	1.00000
,	•	34.6400	45.3600	40.0000	5.3600	00000
~	•	20.6713	25.5124	22.0033		000000
•	•	47			C014.7	0.52037
		0001-19	0+08.76	80.0000	12.6640	0.5000
	•	4.4082	7.5916	6.0000	1.5014	
•		4.0000	0000		0::5::	000000
•			00000		0.0	1.00000
•	•	6.8330	5.4402	3.9934	1.2957	0.55401

SLBNET DGER 3

THE CROER IN WHICH TO CONSIDER THE ACTIVITIES TO DETERBINE THE LCNGEST PATH TIME:

0.949930 02 THE CRITICAL PATH TIME WHEN EACH ACTIVITY'S COMPLETION TIME IS SET EQUAL TO ITS MEAN IS =

PE 8 ACDES CA THE CRITICAL PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL MODE: 8. 7. 6. 5. 4. 3. 2. 1.

THE 7 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY: 13. 14. 6. 7. 9. 11. 4.

A LOWER BOUND ON THE SUBNETHERN DURATION TIME 15 = 0.834970 02

8 NODES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL MODE: 8. 7. 6. 5. 4. 3. 2. 1.

THE 7 ACTIVITIES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY: 13. 14. 6. 7. 9. 11. 4.

A UPFER BOUND CA THE SUBNETHORK DURATION TIME IS # 0.113040 03

6 ACDES CA THE LCAGEST PATH ARE AS FOLLCUS BEGINNING WITH THE TERMINAL MODE:

5 ACTIVITIES CA THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY: 13. 14. 8. 9. 3.

7. 15 = 9. 15 = THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 6-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY 11. IS = . IS . 14. 15 . 6. IS # 1-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 2-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE NUMBER OF ASSOCIATES ASSOCIATED MITH THE 4-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE S-TH CRITICAL PATH ACTIVITY. 1.E. ACTIVITY THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 7-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 3-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FCLLOWS 8. THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FCLLOWS THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS 0.100000 01 THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FCLLODS 2. THE NUMBER OF ASSCCIATES ASSOCIATED WITH THE THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE THE NUMBER OF ASSECTATES ASSOCIATED WITH THE THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBOA .

			S ACTIVITY B IS ACTIVITY C IS ACTIVITY	2 22 2 2 2 2 2
	10 ACTIVITY 10 ACTIVITY 10 ACTIVITY 10 ACTIVITY		10 ACTIVITY 10 ACTIVITY 10 ACTIVITY 10 ACTIVITY	10 ACTIVITY 10 ACTIVITY 10 ACTIVITY 10 ACTIVITY 11 ACTIVITY 11 ACTIVITY 11 ACTIVITY
3 5 6 112 112 NOT THE CRITICAL PATRA.	CORRESPONDING CORRESPONDING CURRESPONDING COFRESPONDING CURRESPONDING	CORRESPONDING CORRESPONDING CORRESPONDING CORRESPONDING	CURRESPONDING CORRESPONDING CORRESPONDING CORRESPONDING CURRESPONDING	CURRESPONDING CORRESPONDING CORRESPONDING CORRESPONDING CORRESPONDING CORRESPONDING CORRESPONDING
8 8 8 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	13 4 ELIMINANTS 1-Th ELIMINANT 2-TH ELIMINANT 3-TH ELIMINANT 4-TH ELIMINANT	3 ELIMINANTS 1-TH ELIMINANT 2-TH ELIMINANT 3-TH ELIMINANT 2 ELIMINANTS	2-TH ELIMINANT O ELIMINANTS 2 ELIMINANTS 1-TH ELIMINANT 2-TH ELIMINANT	6 ELIMINANTS 6 ELIMINANTS 7-TH ELIMINANT 9-TH ELIMINANT 6-TH ELIMINANT 6-TH ELIMINANT 6-TH ELIMINANT

10 000001.0

THERE ARE I NC. EMPTY CLUSTERS AFTER POOLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

THERE ARE 12 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS:

7 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY WERE AS FOLLOWS:

-	
-	
0	
1	
0 000001	
0	
0	
0	
0	
-	
0	
-	
10 000001	
-	
0	
ŏ	
ŏ	
o	
0	
-	
0	
BHEN (THETA.LAMBOA) = (
-	
-	
2	
9	
Ξ.	
-	
•	
-	
•	
	
¥	
÷	
=	
Z	
W	
I	
•	
-	
•	
m	
•	
•	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
"	
BOUNES ON THE DURATION DISTRIBUTION FOR SUBNETWORK 3	

										_						_				
3	-05	-05	-01	10-	10-	-01	-01	8	8	8	°	8	00	00	8		8	8		5
UPPER BOUND	0.232210-02	0.656640-02	0-137510-0	0-182130-0	0-205366-0	0-303740-0	0-803980-0	0.119360	0.188180	0.354090	0.505880	0.523000	0.679620	0.679620	0.753740	0.888840	0.88840	0.866840	0.888840	C. 1 0000D
OND	-05	-05	10-	10-	10-	-01	-01	00	00	00	00	00	00	00	00	00	00	00	00	10
LOWER BOUND	0.232210-02	0.232210-02	0-110110-0	0-182130-0	0-505360-0	0-224583-0	0-751670-01	0.119300	0.188180	0.354090	0.505880	C.52300D	0.673620	0.679620	0.753740	0.888840	0.888840	0.884840	0.848840	G-10000D
	05	0.5	05	0.5	92	05	0.5	63	63	03	03	63	03	03	63	03	03	03	63	03
3.1.	0.856460	0.e7755C	0.895440	0.920920	0.542410	0.963900	0.985360	0.100650	0-102640	0-104580	0-107130	0-107790	0-108450	001501.0	0-109760	0-110420	0.111100	0.111736	C.11239D	0-113040

THE BCLNDS MERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

CALY 540 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS BERE CONSIDERED.

SYSTEMATIC SAMPLING BAS USED.

1-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 13. 15 -THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 2-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 14. IS # 6. 15 = 7. 15 = 9. IS = 11. 15 = +· 15 = THE NUMBER OF ASSCRIATES ASSOCIATED WITH THE T-TH CRITICAL PATH ACTIVITY. 1.E. ACTIVITY THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 3-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 6-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 5-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE NUMBER OF ASSCCIATES ASSOCIATED WITH THE 4-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS 8. THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS 12. THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FCLLCUS 0.200000 01 THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE THE ASSOCIATES ARE NOW ICENTIFIED FOR LANBOA -

THERE ARE S NCHEMPTY CLUSTERS AFTER POQLING CN THE BASIS OF ASSOCIATES ONLY.

13. 12.

THE ACTIVITIES IN THE 2-TH CLUSTER ARE AS FOLLOWS:

14.

THE ACTIVITIES IN THE 3-TH CLUSTER ARE AS FOLLOWS:

THE ACTIVITIES IN THE 5-TH CLUSTER ARE AS FOLLOWS:

9.

THE ACTIVITIES IN THE 6-TH CLUSTER ARE AS FOLLOWS:

11. 2. 4.

THERE ARE 5 ACTIVITIES NOT IN ANY CLUSTER YET.

THERE ARE I NCHEMPTY CLUSTERS AFTER POCLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS. THERE ARE 13 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS:

7 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY WERE AS FOLLOWS:

	TIME		LOWER BOUND	UPPER BOUND	
•	0.856460	0.5	0.232130-02	0.232130-02	
	0-677550	05	0.696380-02	0.696380-02	
•	0.899440	05	0.928500-02	0.928500-02	
•	6.920520	70	10-000011-0	0-116060-01	
•	0102060	0.5	0.249890-01	0.249890-01	
	0.563500	05	0.348610-01	0.348610-01	
	0.585380	05	0.815520-01	0.815520-01	
	0-100690	63	0.131540 00	0.131540 00	
	0.102640	63	0.149280 00	0.189280 00	
	0-104590	03	0.341770 00	0.341770 00	
•	0.107130	63	0.0 00000000	0.499980 00	
	0-167790	63	0.521370 00	0.521370 00	
	0-108450	63	0.647070 00	0.647070 00	
	001501-0	03	0.647070 00	0.647070 00	
	0-105760	63	0.738370 00	0.738370 00	
	0-110420	3	0.684910 00	0.884910 00	
•	0-1111070	03	0.00 0164810	0.884910 00	
•	0.111730	63	0.00 0164010	0.884910 00	
•	0-112390	63	0.684910 00	0.884910 00	
•	0-113340	63	0.1000000 01	0.100000 01	

0.200000 01 1

0.200000 01 .

THE BCUNCS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

CALY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS WERE CONSIDERED.

SYSTEMATIC SAMPLING BAS USED.

13. 15 = 14. 15 = 6. 15 = . 15 = 7. 15 = 1-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 6-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 2-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 3-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 4-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 5-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 7-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FCLLOWS THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS 12. THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS 0.200000 01 THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE THE NUMBER OF ASSCCIATES ASSOCIATED WITH THE THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE THE NUMBER OF ASSOCIATES ASSUCIATED WITH THE THE ASSCCIATES ARE NOW ICENTIFIED FOR LAMBDA .

THERE ARE S NCNEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF ASSOCIATES ONLY.

THE ACTIVITIES IN THE 1-TH CLUSTER ARE AS FOLLOWS:

14.

THE ACTIVITIES IN THE 3-TH CLUSTER ARE AS FOLLOWS:

6. 8. 7.

THE ACTIVITIES IN THE 6-TH CLUSTER ARE AS FOLLOWS:

7.

THE ACTIVITIES IN THE 6-TH CLUSTER ARE AS FOLLOWS:

11. 2. 4.

THERE ARE S ACTIVITIES NOT IN ANY CLUSTER YET.

THERE ARE I NCHEMPTY CLUSTERS AFTER POCLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

THEKE ARE 13 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS:

7 CLUSTERS MAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY BERE AS FCLLOWS:

-	
-	
=	
1.200000 01	
•	
×	
ă	
ŏ	
×	
9	
14	
•	
0	
•	
300000	
0	
0	
0	
0	
0	
0	
-	
0	
-	
-	
-	
~	
×	
#	
-	
٠,	
_	
•	
•	
-	
w	
I	
-	
-	
Z	
E	
HEN	
MHEN	
MHEN	
3 WHEN (THETA.LAMBDA) =	
3 WHEN	
3 WHEN	
S WHEN	
3 WHEN	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
•	
BCUNDS ON THE DURATION DISTRIBUTION FOR SUBNETBORK 3 WHEN	

ONO	-05	-05	20-1	101	10-0	10-	10-	00	00	00	00	00	00	00	00 0	00 0	00	00	00	
UPPER BCUND	0.232130-02	0.656380-02	0.928500-02	0-116060-0	0-249890-01	0-348610-0	0-815520-0	0.131540	0.189280	0.341770	086654.0	0.521376	0.647070	0.647070	0.738370	0.684910	0.684910	0.884910	0.884910	000001-0
ONO	-05	-02	-05	10-	-01	10-	10-	00	00	00	00	00	00	00	00	00	00	00	00	
LOWER BCUND	0.232130-02	0.696380-02	C- 528500-02	0-116060-0	C-24989D-01	0-348610-01	0-075579-0	0.131540	0.189280	0.341770	0.474980	0.521370	0.647070	0.047070	0.738370	0.6489.0	0.834910	0.484910	0.484910	000001
	95	05	02	95	05	05	20	63	03	03	03	03	60	03	63	63	03	63	63	
	0.650400	0.677950	0.89946	0.920920	0.542410	0.963500	0.985380	0.10cesp	0-102640	0.104580	0-107130	0-101190	0-108450	0.1001.0	091531-0	0.110420	0.1111070	0.111730	0.112390	0401110

THE BCUNDS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

CALY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS BERE CONSIDERED.

SYSTEMATIC SAMPLING BAS USED.

THE LFFER AND LOWER BCUNDS ON THE DISCRETE SUBNETWORK DURATION DISTRIBUTION ARE EQUAL.
THUS AN ESTIMATE OF THE CISCRETE DURATION DISTRIBUTION FOR SUBNETWORK 3 IS

DISTRIBUT																					
0	10	20	20	02	10	10	10	10	00	00	00	00	00	00	00	00	00	00	00	00	10
DURATION	STRIBUT	232130-	-08 c 95 o	-00SP	110000-	-0686	348610-		15.0	9280	1770	_	1370	0707	7070	9370	0160	0160	0160	910	0000
CISCAETE	1810	0.23	50.0	0.92	0.11	0.24	0.34	10.0	0.13	0.14	0.34	64.0	0.52	0.04	0.04	0.73	0.84	0.80	0.08	0.88	0.10
CF 17.E		20	05	70	02	02	05	05	63	63	03	3	63	03	03	03	63	63	63	63	03
ESTIMATE C	3111	0.650400	0.877550	0.89940	6.520520	0.942410	C.\$63900	0.985380	0.100090	0.102440	0.104980	0.107130	0.107790	0.108450	0.105160	0.105760	0-110420	0.111370	0.111730	0.112350	0.113040
4																					

THIS IS THE CUTPUT FACE THE SYNTHESIS PROGRAM: SYNTH

THE CLARENT PROJECT SCHEDLLE IS THE ONE MOST RECENTLY LISTED BY THE DETERMINISTIC SCHEDULE RESOLUTION PROGRAM. THE PROJECT'S CORRESPONDING APPROXIMATE COMPLETION TIME DISTRIBUTION IS DETERMINED BELOW:

FCA THIS FACELEM

THE ALWBER CF C.D.F. SUBDIVISIONS USED THROUGHOUT SYNTHESIS IS 49.
THE TARGET THE USED BY THE DEFERMINISTIC SCHEDULER WAS 128.00000
THE WEAN OF THE PROJECT'S APPECATME COMPETION TIME DISTRIBUTION WILL BE COMPARED TO THE SPECIFIED FROJECT DEACLINE TIME OF 125.00000

THE INSTRUCTIONS TO BE PERFORMED ARE

SUBNETWORKS TO BE SYNTHESIZED SERIES=0 PARALLEL=1 • SUBNETBORK INSTRUCTION NO.

1

	•	DETENTINED	200	SUBNETWORK ANALYSIS	ANAL TO
15.6000	CX) 4				
20.0876	0.05000				
23-3455	0.15000				
25.5891	0.25000				
27.4362	0.35000				
29.1490	0.45000				
33.6000	0.55000				
33.6000	0.059.0				
37.2000	0.15000				
37.2000	0.05000				
*0.8000	0.65000				
0000	1.00000				
*	F(x)				
1940.68	0.00232				
87.7948	0.00696				
89.5435	0.00929				
52.0522	0.01161				
64.2409	0.02499				
96.3890	0.03486				
56.5364	0.08155				
1789.001	0.13154				
102-8358	0.18928				
104.5645	0.34177				
107.1332	8666**0				
1C7.7898	0.52137				
108.4463	0.64 707				
169-1029	0.64707				
105.7594	0.73837				
110.4160	0.88491				
111.0725	0.86491				
111.7291	0.88491				
112.3856	0.88491				
113.0422	1.00000				

****	0.0	0.0	9	0.0	6.00012	6.00012	0.00035	6.000.3	0.00010	0.00139	0.00151	0.00200	6.00334	0.00427	6.60522	0.00703	0.01029	0.01262	0.02228	0.0250	0.0500	0.06476	0.07313	0.11432	0.12435	0.17084	0.20357	0.24976	3466	0.41044	0.45992	0.51173	0.54260	0.58624	0.66265	0.88903	0.74598	0.79715	C. 83843	0.88644	0.50878	C.54053	0.90349	C. 57084	0.98235	0.99425	0.55425	1.00000
	HILLIAMES	HELP BREAK	1131,83983	1104.6888.4	105.63288	1581661	118-12727	129.47813	110.82544	111.54785	110.514.51	113.64187	111.400.5	110.19924	117.36211	13500.811		74.700			118.116.11	120.07703	117.62365	144.77675	154.91761	171.00.07	132.21133	133.35615	131.00	136.79876	137.54562	135.05248	140.23534	141.38620	142.53306	143.67552	144.82678	145.57364	147.12050	144.26736	149.41422	150-56108	151.70794	152.85460	154.00166	155,14652	156.29538	157.44224

THE STATES IN THE STATES LZED NETBORK IS

THE CIFFERENCE BETWEEN 139.41685 AND THE PROJECT DEADLINE IS PENCE. THE NEW TARGET TIME IS 112.07397 THE MEAN OF THE SYNTHESIZED DISTRIBUTION IS 139.41685

11.53 PERCENT OF THE PROJECT DEADLINE.

THIS COMPLETES ITERATION

THIS IS THE QUIFUT FROM THE CETERMINISTIC SCHEDULE RESOLUTION PROGRAM: DSR

••
S
IS AS FOLLOWS
0
_
٠.
_
0
=
v,
•
-
S
_
_
•
0
-
.,
-
•
12.07396
-
-
_
LIME O
_
ш
=
-
_
_
_
_
w
œ
*
ARGET
TAR
TAR
TAR
A TAR
-
FOR A TAR
FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
SCHEDULE FOR A T
FOR A T
PRCJECT SCHEDULE FOR A T
PRCJECT SCHEDULE FOR A T
PRCJECT SCHEDULE FOR A T
PRCJECT SCHEDULE FOR A T
PRCJECT SCHEDULE FOR A T
SCHEDULE FOR A T
PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T
PRCJECT SCHEDULE FOR A T
WINING CCST PRCJECT SCHEDULE FOR A T
WINING CCST PRCJECT SCHEDULE FOR A T
WINING CCST PRCJECT SCHEDULE FOR A T
WINING CCST PRCJECT SCHEDULE FOR A T
WINING CCST PRCJECT SCHEDULE FOR A T
CCST PRCJECT SCHEDULE FOR A T

4C11v1TV:	FEAN	DISTRIBUTION	PARAMETERS	ERS						
	19.07396	NGRMAL	PEANE	19.07396. ST.DEV=	ST.DEV=	5.44970				
~	14.00000	NURMAL	MEANS	14.00000.	ST.06 V=	3.61616				
•	10.0000	FIXECICCUSTANT	TIPE	10.00000						
	20.00000	NCRMAL	MEANE	10.00000	ST.05V	11.66666				
•	39.00000	NCFMAL	MEANE	39.00000. SY.DEV=	SY.DEV=	4.59619				
•	32.00000	BETA	= 2] =	12.30770. MAXE	MAX=	61.53848. ALPHA=	ALPHA=	0.79428. BETA	BETA	1.69143
	13.00000	NORMAL	MEANE	13.00000.	ST.DEV.	2.14967				
	34.00000	NCEMAL	MEANE	34.00000.	ST.DEV=	3.13846				
•	2.00000	FIXEDICCUSTANT	TIME	2.00000						
01	21.00000	BETA	II Z	9.54546.	MAX=	38-18184. ALPHA=	ALPHA=	12.99998. BETA=	BETA	200002
=	10.0000	FIXEC(CONSTANT)	TIME	10.00000						
12	10.00000	NORMAL	MEANE	10.00000	ST.DEV=	2.79508				
2	990000	NCFWAL	ME ANS	.00000.	ST.DEVE	9.83869				
•	40.0000	NGRMAL	MEANE	.00000.0	ST.DEV=	5.71428				
==	80.0000	NCEWAL	MEANE	80.00000. ST.DEV=	ST.DEV=	13.71428				
16	4.00000	FIXECICONSTANT	TIME	♦.00000						
11	4.00000	BETA	HINE	1.33333.	MAX=	8.00000. ALPHA=	ALPHA=	1.00000. BETA.	BETAS	2.00000
10	000000	NOFWAL	MEANS	6.00000. ST.DEV=	ST.DEV=	1.69705				

ACTIVITY	CRIGIN	TERMINAL 2	NAL MEAN	STANDARD DEVIATION 5-13
		-		
		0.0	10.000	
		1 50.0	12.7844	
		2000	16-6172	
		0.35	17.8740	
		0.45	18.9984	
		0.55	22.7115	
		0.00	25.2538	
		0.85	25.2538	
		0.95	30.3384	
		1-00-1	32.8808	
ACTIVITY	ORIGIN 2	TERMINAL 6	NAL MEAN	STANDARD DEVLATION
		50.		
		1 000	35.0000	
		0.05	50.8092	
		0-15	57.9081	
		0.25	62-1310	
		0.45	68.5340	
		0.55	71.4660	
		0.65	74.4554	
		0.75	77.8690	
		0.95	89-1908	
		1.00	105.0000	

ACTIVITY		TERMINAL	MEAN MEAN	STANDARD DEVLATION
•	n	•	34.0	2.94
		613		
		0.0	24.5846	
		- 50.0	28.8375	
		0-15	30-7471	
		0.25	31.8831	
		0.35	32.7907	
		0.45	33.6056	
		0.55	34.3944	
		0.65	35.2093	
		0.75	36.1169	
		0.85	37.2529	
		-	39.1625	
			43.4154	
ACTIVITY	M10100	TEDALINA	77.	ACTA TO SO COLOTA
•	•	•		0000
		200		
		0.0	5.0000	
			5.0000	
		0-15	2.0000	
		0.25	5.0000	
		0.35 -	2.0000	
		0.45	2.0000	
		0.55	2.0000	
		0.65	2.0000	
		0.75	2.0000	
			2.0000	
			5.0000	
		1.00	2.0000	
ACTIVITY	CRIGIN	TERMINAL	AL MEAN	STANDARD CEVEATION
01	F	•		2.21
		0.0	9.5455	
			17.2433	
		0.15	18.5502	
		0.25	19.3660	
		0.35	20.0346	
		0.45	20.6459	
		0.55	21.2456	
		0.65	21-8719	
		0.75	22.5750	
		0.85	23.4600	
		0.95	24.9458	
		1-00	38-1818	

ACTIVITY ORIGIN TERMINAL MEAN STANDARD DEVIATION 44.0 9.23 ACTIVITY CRIGIN TERMINAL MEAN STANDARD DEVIATION 44.0 9.23 ACTIVITY CRIGIN TERMINAL MEAN STANDARD DEVIATION 9.23 ACTIVITY STANDARD DEVIATION 9.23 ACTIVITY CRIGIN TERMINAL MEAN STANDARD DEVIATION 9.23	ACTIVITY	CA 16 IN	TERMINAL	MEAN 10.0	STANDARD DEVIATION
ORIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN S			500		
ORIGIN CRIGIN CRIGIN			0.0	10.0000	
ORIGIN CRIGIN CRIGIN			0.05	10.0000	
ORIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN			0.15	10.0000	
ORIGIN CRIGIN CRIGIN			0.25	10-0000	
ORIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN S			0.35	10.0000	
ORIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN			0.45	10.0000	
CRIGIN TERMINAL MEAN S			0.55	10.000	
ORIGIN TERMINAL MEAN ORIGIN TERMINAL MEAN O.05 10.0000 O.05 10.0000 O.05 10.0000 O.05 10.000			0.65	10.0000	
ORIGIN TERMINAL MEAN FITT 1 O.05 10.0000 1.00 10.0000 O.05 10.001 O.05 10.01 O			0.75	10.0000	
CRIGIN TERMINAL MEAN S 7 10.000 0.35 10.050 0.45 9.6488 0.55 10.051 0.65 11.6453 0.65 12.6970 0.75 11.6453 0.75 14.4639 0.75 14.4639 0.75 14.4639 0.75 14.4639 0.75 14.4639 0.75 14.4639 0.75 14.4639 0.75 14.4639 0.75 14.4639 0.75 14.7910 0.75 42.7937 0.45 42.7937			0.85	10.0000	
CRIGIN TERMINAL MEAN 6.05 1.6147 6.05 5.6023 6.15 7.1030 6.25 10.3312 6.45 9.6488 6.25 10.3312 6.45 9.6488 6.25 11.8853 6.45 11.8853 6.45 12.8970 6.45 12.8970 6.45 14.5977 1.00 14.4839 6.05 14.7910 6.75 17.910			0.95	10.0000	
CRIGIN TERMINAL MEAN 6.05 1.40147 6.05 5.4023 6.15 7.4030 6.25 10.3512 6.45 9.4648 6.25 11.3853 6.45 11.8853 6.45 11.8853 6.45 11.8853 6.45 11.8853 6.45 11.8853 6.45 11.8853 6.45 14.5977 1.00 14.4839 6.25 14.5977 6.25 14.5937 6.25 14.5937 6.25 14.5937 6.25 14.2039 6.25 14.2039 6.25 40.2090 6.25 40.2090 6.25 47.7910 6.25 47.7910 6.25 47.7910 6.25 60.1840 6.25 60.1840			1-00	10.0000	
CRIGIN TERMINAL MEAN FILT 1 0.0					
CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN S 7 100-0 10-05 5-4023 0-45 9-6488 0-45 9-6488 0-45 10-3512 0-65 11-6457 0-75 11-6453 0-75 11-6453 0-75 12-6970 1-00 14-4639 0-15 14-4639 0-16 14-4639 0-17 14-4639 0-18 14-4639	***************************************	200	TEGAL		MOTTAL STATE OF THE PARTY
FITTION TO THE T	12	4	-		2.62
CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN FITT T 44.0 FITT T 7 44.0 CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN FITT T 7 44.0 CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN FITT T 7 44.0 FITT T 7 7 74.0 CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN FITT T 7 7 74.0 CRIGIN TERMINAL MEAN FITT T 7 7 74.0 CRIGIN TERMINAL MEAN FITT T 7 7 74.0 CRIGIN TERMINAL MEAN FITT T 7 7 7 74.0 CRIGIN TERMINAL MEAN FITT T 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	:				
CRIGIN CRIGIN			FITT		
CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN F(T 1) T 44.0 CRIGIN TERMINAL MEAN S 7 10.20 CRIGIN TERMINAL MEAN S 7 27.8160 CRIGIN TERMINAL MEAN F(T 1) T 44.0 F(T 2) T 44.0 F(T 3) T 44.0 F(T 3) T 5.8160 CRIGIN TERMINAL MEAN S 7 7 7 910 CRIGIN TERMINAL MEAN S 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8				1.0147	
CRIGIN TERMINAL MEAN S 1.0030 0.45 9.6488 0.45 9.6488 0.45 10.3312 0.45 11.8853 0.45 12.8970 0.45 12.8970 0.45 12.8970 0.45 14.5977 0.65 14.5977 0.45 14.5977 0.45 14.5977 0.45 14.4839 0.			0.05	5.4023	
CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN FITTO 0.45 10.4512 0.45 11.4653 0.45 11.4653 1.00 14.4639 0.00 14.4639 0.00 14.4639 0.05 27.4860 0.15 33.4639 0.45 45.2043 0.45 45.2043 0.45 45.2043 0.45 50.6041 0.45 50.6041 0.45 60.1840			0.15	7.1030	
CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN FAT J 0.05 10.3512 0.05 11.0070 0.05 14.5977 1.00 18.3653 FAT J 0.0 18.3653 0.05 14.4839 0.05 27.860 0.15 37.3639 0.25 42.7637 0.45 42.7637 0.45 42.7637 0.45 42.7637 0.45 42.7637 0.45 42.7637 0.45 42.7637 0.45 42.7637 0.45 60.1840 1.00 73.5161			0-25	6-1147	
CRIGIN TERMINAL MEAN FILT) FILT) FALT) F			0.35	8.9230	
CRIGIN TERMINAL MEAN CRIGIN TERMINAL MEAN FIT T 0.05 10.355.3 1.00 18.3653 FIT T 0.0 14.4639 0.05 27.8160 0.15 33.8027 0.25 42.7637 0.45 42.7637 0.45 42.7637 0.45 42.7637 0.45 60.1840 1.00 73.5161			0.45	9.6488	
CRIGIN TERMINAL MEAN FAT 1 0.05 11.0853 0.05 14.5977 1.00 18.3853 FAT 1 0.0 18.3853 0.05 27.8860 0.05 27.8860 0.05 27.8860 0.05 27.8060 0.05 42.7637 0.45 42.7637 0.45 42.7637 0.45 42.7637 0.45 60.1840 1.00 73.5161			0.55	10.3512	
CRIGIN TERMINAL MEAN 5 F(T) 7 44.0 F(T) 7 44.0 F(T) 7 44.0 F(T) 7 44.0 F(T) 7 7 44.0 F(T) 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7			0.65	11.0770	
CRIGIN TERMINAL MEAN 5 FILT T 44.0 FILT T 44.0 6.05 12.6970 0.05 14.4839 0.05 27.6160 0.15 33.6027 0.45 45.2363 0.45 45.2363 0.45 50.6361 0.45 50.6361 0.45 60.1840			0.75	11.8853	
CRIGIN TERMINAL MEAN 5 F(T) T 44.0 6.0 - 14.4839 0.05 27.8160 0.15 33.8027 0.25 42.7637 0.45 42.7637 0.45 42.7637 0.45 50.204 0.45 60.1840 1.00 73.5161			0.85	12.8970	
CRIGIN TERMINAL MEAN FAT J 0.0			0.95	14.5977	
CRIGIN TERMINAL MEAN 5 7 44.0 7 6(1) T 6.0 6.05 14.4639 6.05 33.6027 6.25 37.8029 6.35 42.7637 6.45 42.7637 6.45 42.7637 6.45 42.7637 6.45 42.7637 6.45 60.1840 1.00 73.5161			1.00	18.3853	
FILT TERMINAL MEAN FILT T 0.05 14.4839 0.05 14.4839 0.05 33.6027 0.45 37.3039 0.45 40.2090 0.45 40.2090 0.45 40.7037 0.45 50.401 0.45 50.401 0.45 50.401 0.45 50.401 0.45 60.1973					
F(T) T + 44.0 F(T) T + 44.0 F(T) T + 44.0 0.0 14.4639 0.05 27.6160 0.15 33.6027 0.25 42.7637 0.45 42.7637 0.45 45.2363 0.45 50.6161 0.45 60.1840					
F(T) T T	ACTIVITY	CRIGIN	TERMI		STANDARD DEVIATION
	13	'n	2	***	9.23
111111111111			F(T)		
11111111111			1 000	14.4839	
1111111111			0.05	27.8160	
111111111		,	0.15	33.8027	
11111111			0.25	37.3639	
1111111			0.35	40.2090	
111111			0.45	42.7637	
11111			0.55	45.2363	
1111			0.65	47.7910	
111			0.75	50.6361	
11			0.85	54-1973	
1			0.95	60-1840	
			1.00	73.5161	

		40.0	40.0	9
	£(1)			
	0.0	22.8572		
	0.00	30.6004		
	0.15	34.0774		
	0.25	36-1458		
		37.7982		
	0.55	40-7181		
	0.65	42-2018		
	0.75	43.8542		
	0.85	45.9226		
	0.95	49.3596		
	1.00	57.1428		
ORIGIN	TERMI			DEVIATION
7				9
	F(T)	1		
	0.0	36.8572		
	0.05	57.4410		
	0.15	65.7858		
	0.25	10-7499		
	0.35	74.7156		
	0.45	78.2767		
	0.55	81.7233		
		85.2844		
		89.2501		
		94.2142		
	1	0666-201		
	1	97410171		
CRIGIN	TERMI			CENTATION
1				0
		•		
	0.0	4.0000		
	0.05	**0000		
	0.15	4.0000		
	0.25	4.0000		
	0.35	**0000		
	0.45	** 0000		
	0.55	4.0000		
	0.65	4.0000		
	0.75	4.0000		
	0.85 -	4.0000		
		4.0000		
		4.0000		
	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		TERMINAL TERMIN	0.05 22.8572 0.05 30.0004 0.25 30.1458 0.35 37.7982 0.65 40.7181 0.65 40.7181 0.65 45.926 0.95 45.926 0.95 45.926 0.05 57.1428 0.05 57.1428 0.05 57.1428 0.05 57.1428 0.05 57.1428 0.05 57.1428 0.05 57.1428 0.05 57.1428 0.05 57.1428 0.05 65.9261 0.05 65.261 0.05 65.261 0.05 65.261 0.05 60000 0.05 60000 0.05 400000 0.05 400000

		LEKHINAL		STANDARD CEVIATION
-		•	•••	1.30
		50.13		
		0.0	1.3333	
		0.05	1.9840	
		0.15	2.5292	
		0.25	2.9535	
		0.35	3.3398	
		0.45	3.7155	
		0.55	4.0979	
		0.65	4.5038	
		0.75	4.9579	
		0.65	5.5103	
		0.95	6.3427	
		1.00	8.0000	
ACTIVITY	CRIGIN	TERMINAL	AL MEAN	STANDARD CEVIATION
=	•	•		1.59
		F(T)	-	
		0.0	0.9088	
		0.05	3.2085	
		0.15	4.2411	
		0.25	4.8554	
		0-35	5.3461	
		0.45	5.7867	
		0.55	6.2133	
		0.65	6.6539	
		0.75	7.1446	
		0.85	7.7589	
		0.95	8.7915	
		1.00	11.0912	

THIS IS THE CUTPUT FREM THE SUBNETWORK ANALYSIS PROGRAM: SUBNET

EACH SUBNETLICER'S AFPROXIMATE CURATION DISTRIBUTION IS DETERMINED.

SLUNETOCAK

MEAN STANDARD DEVIATION PROB. LOWER PT. 20.7743 5.1270 TERRINAL LOSER POINT UPPER POINT 2 16-1595 26-4703 ACTIVITY ORIGIN INITIAL INPUT

~													
DAK	NO		5	00	00	0	00	00	°	00	00	00	5
SUBNET	DISTRIBUTION	0.0	0-200000-0	0.150000	0.250000	0.350000	0.450000	C.55000D	0.650000	C.75000D	0.00084-0	0.95000	0-100000
FOR	٠	۰	۰	•	٥	٠	•	٠	•	•	•	•	•
IBUTICA		02	02	05	0.5	05	05	0.5	0.5	05	05	95	05
DISTR		300001-0	0.127640	0.152040	166130	0-176740	0.1essec	0-227120	227125	52540	0.252540	0.303360	0.328810
THE CLEATICH DISTRIBUTION FOR SUBNETWORK	1146	0.10	0.1.0	0.1	0.10	0.1	0.10	0.2	0.2	0.23	6.25	0.3	0.32
17.6													

		VIATION PROB. LOWER PT.														
		STANDARD DEVIATION	4.311	10.033	2.016	0.0	2.621	2.943	9.228	0.0	5.360	2.208	12.864	1.591	0.0	1.295
		MEAN 70.0000	39.0000	31.9497	13.0000	10-0000	10.0000	34.0000	***0000	2.0000	*0.000	20.9938	80.0000	000009	**0000	2.9934
		UPPER POINT 80.9433	43.3112	43.2217	15.0164	10.000	12.6218	36.9439	53.2287	2.0000	45.3600	23.2939	92.8640	7.5918	**0000	5.4402
		LOWER POINT 59.0567	34.6888	23.0185	10.9836	10.000	7.3782	31.0561	34.7713	2.0000	34.6400	18.8738	67.1360	4.4082	4.0000	2.6330
		TERMINAL		•	~		•	•	•	•	•	•	•	•		•
•	NPLT	ORIGIN	-		-	s	s	•	•	•	•	2	~		•	•
SUBNETECER	INITIAL INPLT	ACTIVITY	~	•	•	•	•		•	•	•	:	12	13	:	15

THE CREER IN WHICH TO CONSIDER THE ACTIVITIES TO DETERMINE THE LONGEST PATH TIME:

0.930000 02 THE CRITICAL PATH TIME WHEN EACH ACTIVITY'S COMPLETION TIME IS SET EQUAL TO ITS MEAN IS .

THE 3 NCDES CN THE CRITICAL PATH ARE AS FOLLOWS BEGINNING BITH THE TERMINAL NODE:

8. 2. 1.

THE 2 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING BITH THE TERMINAL ACTIVITY:
12. 4.

A LOWER BOUND ON THE SUBNETWORK DURATION TIME IS = 0.817000 02

THE 8 NODES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL NUDE: 8. 7. 6. 5. 4. 3. 2. 1.

7 ACTIVITIES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:

A UPFER BCUND ON THE SUENETWORK DURATION TIME IS # 0.11304D 03

E 6 NCDES CN THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL NODE: 8. 7. 6. 4. 3. 1.

5 ACTIVITIES CN THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY: 13. 14. 8. 9. 3.

TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/6 9/2
A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJEC--ETC(U)
AUG 78 T C BAKER, R L SIELKEN N00014-78-C-0426 AD-AU60 568 THEMIS-TR-57 UNCLASSIFIED NL 2 OF 4 AD A080568 1

THE ASSECTATES ARE NOW IDENTIFIED FOR LAMBDA . 0.100000 01

THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 1-TH CRITICAL PATH ACTIVITY. 1.E. ACTIVITY 12. IS

THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS 13. 14. 6. 7. 2.

THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 2-TH CRITICAL PATH ACTIVITY. 1.E. ACTIVITY 4. IS a

THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS

THERE ARE I NCHEMPTY CLUSTERS AFTEK POGLING CN THE BASIS OF ASSOCIATES CHLY.

THE ACTIVITIES IN THE 1-TH CLUSTER ARE AS FOLLOWS: 12. 13. 14. 6. 7. 2. 4.

TPERE ARE 8 ACTIVITIES NOT IN ANY CLUSTER YET.

:sac	2 •	2 ◆	21 *	2.	21 *	21 •		21	2 •	
THEY ARE AS FOLLOWS:	IS ACTIVITY IS ACTIVITY	ACTIVITY ACTIVITY	ACTIVITY	IS ACTIVITY IS ACTIVITY	IS ACTIVITY IS ACTIVITY	ACTIVITY		IS ACTIVITY	IS ACTIVITY IS ACTIVITY	
*	15	15	5 5	15	15	15		Is	15	
Ĭ.		N N N	m n m	n 000	~~~	00 00	• 9	==	222	•
CAL PATH.	ACTIVITY ACTIVITY ACTIVITY	ACTIVITY ACTIVITY ACTIVITY		ACTIVITY ACTIVITY ACTIVITY	ACTIVITY ACTIVITY ACTIVITY	ACTIVITY ACTIVITY ACTIVITY	ACTIVITY	ACTIVITY	TO ACTIVITY TO ACTIVITY TO ACTIVITY	ACTIVITY
2	222		555	5 555	555	555	5 5	55	555	2
ACTIVITIES NOT ON THE CRITICAL PATH-	2 ELIPINANTS CORRESPONDING TO 1-TH ELIPINANT CORRESPONDING TO 2-TH ELIPINANT CORRESPONDING TO	2 ELIMINANTS CORRESPONDING TO 1-TH ELIMINANT CORRESPONDING TO 2-TH ELIMINANT CORRESPONDING TO	2 ELIMINANTS CORRESPONDING 1-TH ELIMINANT CORRESPONDING 2-TH ELIMINANT CORRESPONDING	0 ELIMINANTS CORRESPONDING TO 2 ELIMINANTS CORRESPONDING TO 1-TF ELIMINANT CORRESPONDING TO 2-TH ELIMINANT CORRESPONDING TO	2 ELIMINANTS CORRESPONDING TO 1-TH ELIMINANT CORRESPONDING TO 2-TH ELIMINANT CURRESPONDING TO	2 ELIMINANTS CORRESPONDING TC ACTIVITY 1-TH ELIMINANT CORRESPONDING TO ACTIVITY 2-TH ELIMINANT CORRESPONDING TO ACTIVITY	O ELIMINANTS CORRESPONDING TO ACTIVITY O ELIMINANTS CORRESPONDING TO ACTIVITY	I ELIMINANTS CORNESPONDING	EL IMINANTS CCARESPONDING EL IMINANT CORRESPONDING EL IMINANT CORRESPONDING	O ELIMINANTS CURRESPONDING TO ACTIVITY
13 ACTIVITIES 2 3 3 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6	2 ELIPINANTS 1-TH ELIPINANT 2-TH ELIPINANT	2 ELIMINANTS 1-TH ELIMINANT 2-TH ELIMINANT	2 ELIMINANTS 1-TH ELIMINANT 2-TH ELIMINANT	2 ELIMINANTS 1-TF ELIMINANT 2-TH ELIMINANT	2 ELIPINANTS 1-TF ELIMINANT 2-TF ELIMINANT	2 ELIBINANTS 1-TH ELIBINANT 2-TH ELIBINANT	O ELIPINANTS	1 ELIMINANTS	2 ELIPINANTS 1-TH ELIPINANT 2-TH ELIPINANT	O ELIMINANTS
THERE ARE 13 ACTIVITIES NOT ON THE CRITICAL PATH. THEY ARE AS FOLLOWS: 2. 2 3. 3 4. 5 6. 6 7. 8 7. 8 9. 10 10. 11		THERE ARE	THERE ARE	THERE ARE THE THE	THERE ARE	7764 ARE THE THE	THERE ARE	THEFE ARE	THERE ARE THE THE THE	THERE ARE

THERE ARE I NCHEMPTY CLUSTERS AFTER PUBLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS. THERE ARE 11 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS: 12. 13. 14. 6. 7. 2. 4. 1. 3. 8. 11.

2 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY BERE AS FOLLOWS:

• 10 000001 • 0																					
(THETA.LAMBDA) = (
BAET BORK 3 BHEN	UPPER BOUND	0.232240-02	0.232240-02	0.464470-02	0.162150-01	0.226780-01	0.431390-01	0.902970-01	0.134790 00	0.203360 00	0.383470 00	0.508140 00	0.508140 00	0.750450 00	0.750490 00	0.179060 00	0.903950 00	0.903950 00	0.903950 00	0.903950 00	0.1000000 01
CISTA IBUTION FOR SU	LOWER BOUND	0.232240-02	0.232240-02	0.404470-02	0-139340-01	0.226740-01	0.300070-01	0.902970-01	0.129410 00	0.203360 00	0.383470 00	0.508140 00	0.504140 00	0. 750490 00	0.750490 00	0.773060 00	0.903950 00	0.903950 00	0.903950 00	C.903v5D 00	10 000001-0
CONTROL OF THE GURATION CISTRIBOTION TOR SUCRETEORS IN CHECK (TRETA-LANGUAL H C	7116	0.634790 02	0.802590 02	0.685380 02	0.508180 02	C.93057C 02	0.953770 02	0.976560 62	0.999350 02	6.102210 03	0-104490 03	0.100775 03	0-107470 63	0.108170 03	C.1 CE 860 03	0.105560 03	0.11026C C3	0-110950 03	0.111050 03	0-112350 03	0.113040 03

0.10000001.0

THE BCUADS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

CMLY 500 OF THE POSSIBLE ACTIVITY BURATION CONFIGURATIONS WERE CONSIDERED. SYSTEMATIC SAMPLING WAS USEU.

THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 1-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY 12. IS ... 0.200000 01 THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS 13. 14. 6. 7. 2. THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBDA .

4. 15 = THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 2-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE ACTIVITIES IN THE ASSUCIATE GROUP ARE AS FOLLOWS 13. 14. 6. 7. 2.

THERE ARE I NCHEMPTY CLUSTERS AFTER POCLING ON THE BASIS OF ASSOCIATES ONLY.

THE ACTIVITIES IN THE 1-TH CLUSTER ARE AS FOLLOWS: 12. 13. 14. 6. 7. 2. 4.

THERE ARE B ACTIVITIES NOT IN ANY CLUSTER YET.

	2.4	21 •	2 ◆		2.4	24	2 +		12	12	~	
THEY ARE AS FOLLOWS:	IS ACTIVITY IS ACTIVITY	ACTIVITY	IS ACTIVITY IS ACTIVITY		ACTIVITY	ACTIVITY	IS ACTIVITY IS ACTIVITY		ACTIVITY	ACTIVITY	IS ACTIVITY IS ACTIVITY	
*	15 A	15 A	15 A		15 4	15 A 21	15 4		IS A	15 A		
T .		200	m m m	9	•••	~~~	60 00 00	•	22	==	13 13 IS 13 IS	•
AL PATH.	ACTIVITY ACTIVITY ACTIVITY	ACTIVITY ACTIVITY ACTIVITY	ACTIVITY ACTIVITY ACTIVITY	ACTIVITY	ACTIVITY ACTIVITY ACTIVITY	ACTIVITY ACTIVITY ACTIVITY	ACTIVITY ACTIVITY ACTIVITY	ACTIVITY	ACTIVITY	ACTIVITY	ACTIVITY ACTIVITY	ACTIVITY
¥ .	555	555	555	2		200	555	10	55	10	200	
ACTIVITIES NOT ON THE CRITICAL PATH- 2 3 5 6 6 7 7 7 10 11 13	CORRESPONDING CORRESPONDING CORRESPONDING	CORRESPONDING CORRESPONDING CORRESPONDING	CORRESPONDING CCHRESPONDING CORRESPONDING	O ELIMINANTS CORRESPONDING TO ACTIVITY	CORRESPONDING CORRESPONDING CURRESPONDING	CORRESPONDING CORRESPONDING CORRESPONDING	CORRESPONDING CORRESPONDING CORRESPONDING	O ELIMINANTS CORRESPONDING TO ACTIVITY	CORRESPONDING CURRESPONDING	CURRESPONDING CURRESPONDING	CURRESPONDING CORRESPONDING CURRESPONDING	O ELIMINANIS CORRESPONDING TO
13 ACTIVITIES 2 3 3 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6	2 ELIMINANTS 1-TH ELIMINANT 2-TH ELIMINANT	2 ELIPINANTS CORRESPONDING 1-TH ELIMINANT CORRESPONDING 2-TH ELIMINANT CORRESPONDING	2 ELIMINANTS CORRESPONDING 1-TH ELIMINANT CCHRESPONDING 2-TH ELIMINANT CORRESPONDING	O ELIMINANTS	2 ELIPINANTS CORRESPONDING TO 1-TH ELIPINANT CORRESPONDING TC 2-TH ELIPINANT CURRESPONDING TO	2 ELIMINANIS CORRESPONDING 1-TH ELIMINANI CORRESPONDING 2-TH ELIMINANI CORRESPONDING	2 ELIMINANTS CORRESPONDING 1-Th ELIMINANT COMRESPONDING 2-Th ELIMINANT CURRESPONDING	O ELIMINANTS	1 ELIMINANTS 1-TH ELIMINANT	1 ELIMINANTS 1-TH ELIMINANT	2 ELIMINANTS CURRESPONDING TO 1-TH ELIMINANT CURRESPONDING TO 2-TH ELIMINANT CURRESPONDING TO	O ELIPINANIS
11-ERE ARE 13-5-5-5-5-5-5-5-5-5-5-5-5-5-5-5-5-5-5-5	TPERE ARE THE TPE	THERE ARE THE THE	THERE ARE THE THE	THERE ARE	THERE ARE	THERE ARE THE THE	THERE ARE THE THE	THERE ARE	TPERE ARE	TPERE ARE	THERE ARE THE THE	TPERE ARE

THERE ARE I NCHEMPIY CLUSTERS AFTER POCLING CN THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

.

TPERE ARE 12 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS:

2 CLUSTERS MAVE BEEN PUBLED TO MAKE THIS CLUSTER. THEY WERE AS FOLLODS:

Q	02	02	02	10	01	10	10	00	00	00	00	00	00	00	00	00	00	00	00	
UPPER BOUND	0.231780-02	0.231780-02	0.463555-02	0.203160-01	0.226335-01	0.352140-01	0.910310-01	0.137910 00	0.216700	0.375800	0.510850	0.510850	0.764810	0.764810	0.780910	0.908240	0.908240	0.908240	0.908240	
LOWER BOUND	0.231780-02	0.231780-02	0.403550-02	0.203160-01	0.226330-01	0.354140-01	0.910310-01	0. 0137910	0.210700 00	0.375800 00	0.510450 00	0.510450 00	00 0184020	00.704810 00	0. 740910 00	0.908240 00	0.908240 00	0.908240 00	0.908240 00	
7114	0.639790 02	0.862550 02	0.665360 02	0.908180 62	C. 53C\$7C 02	0.553770 02	0.970560 62	0.699350 02	0.162216 03	0.104450 03	0-136775 03	6-107-10 03	0.106170 03	0.106865 03	0.105560 63	0.110260 03	6.116650 03	0.111650 03	0.112350 03	

0.200000 01 1

0.20000D 01 ·

THE BCLNDS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

CALY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS WERE CONSIDERED.

SYSTEMATIC SAMPLING BAS USED.

THE ASSOCIATES ARE NOW ICENTIFIED FOR LAMBOA = 0.200000 01

THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 1-TH CRITICAL PATH ACTIVITY. 1.6. ACTIVITY 12. 15 = THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS 13. 14. C. 7. 2.

* 15 * THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE 2-TH CRITICAL PATH ACTIVITY. I.E. ACTIVITY THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FCLLONS

THERE ARE I NCHEMPTY CLUSTERS AFTER POOLING CH THE BASIS OF ASSOCIATES UNLY.

THE ACTIVITIES IN THE 1-TH CLUSTER ARE AS FOLLOWS:

THERE ARE 8 ACTIVITIES NUT IN ANY CLUSTER YET.

There ARE	13 ACTIVITIES NOT ON THE CRITICAL PATH.	NOT ON THE CR	TICAL PA		Ě	THEY ARE AS FULLUES:		;	
'n									
:									
: :									
10.									
:									
2 :	* 51								
THERE ARE			TC ACTIVITY						
THE	1-TH ELININANT		2				11	15	
1	Z-TH ELIMINANT	CORRESPONDING	IO ACTIVITY		12	ACHIVITA			
THERE ARE	2 ELIWINANTS	CURNE SPONDING	TO ACTIVITY		2				
1	I-TH ELIMINAN	CORRESPONDING				IS ACTIVITY		2	
¥.	Z-IF ELIMINANI	CORRESPONDING	IC ACTIVITY		2 13	ACHAIN			
THERE ARE	2 ELIBINANTS	COMMESPONDING	TO ACTIVITY		9				
116	1-TH ELIMINANT	CORRESPOND I NG	10				111	15	
17.5	2-TH ELIMINANT CURRESPONDING	CURRESPONDING	TO ACTIVITY		3 15	ACTIVITY	117		
THERE ARE		O ELIWINANTS CURRESPONDING TO ACTIVITY	TG ACTIV		s				
100 100	2 FUTETABATE	SWIGNOGS SEGO	ALIALIA OL		4				
THE	1-TH ELIMINANT	CURRESPONDING			6 15	ACTIVITY	ITY	12	
115	2-TH ELIMINANT CORRESPONDING	CORRESPONDING	10			ACTIVITY	/1TY	•	
		5.0000000000000000000000000000000000000			,				
THE	1-TH ELIMINANT CURRESPONDING	CURNE SPOND ING	20		15	ACTIVITY	/ITY	12	
146	2-TH ELIMINANT CORRESPONDING	CORRESPONDING	2		7 15		111	•	
These Age		SWIGNOOS SOOD	2						
THE	1-TH ELIMINANT CORRESPONDING	CURRESPUNDING	20		8 15		1111	12	
116	2-TH ELIMINANT CORRESPONDING	CORRESPONDING	TC ACTIVITY		8 15	ACTIVITY	117		
THERE ARE		O ELIPINANTS CORRESPONDING	TO ACTIVITY		•				
THERE ARE	1 ELIPINANTS	CORRESPONDING CORRESPONDING	TO ACTIVITY TO ACTIVITY	177 177 10	5	ACTIVITY	11	12	
THERE ARE THE	I ELIMINANTS	CORRESPONDING CORRESPONDING	TO ACTIVITY TO ACTIVITY	== ££	15	ACTIVITY	E	2	
THERE ARE	2 ELIPINANTS CORRESPONDING	CORRESPONDING	TC ACTIVITY	ITY 13					
115	2-TH ELIMINANT	CURNESPONDING CURRESPONDING	25			IS ACTIVITY	11	·	
TPERE ARE		O ELIMINANTS CONRESPONDING TO	TC ACTIVITY	11v 14					
THERE ARE		O ELIBINANTS CURRESPONDING TO ACTIVITY	TO ACTIV	11v 15					
-									

THEKE ARE I NONEMPTY CLUSTERS AFTER POCLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

THERE ARE 12 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS:

2 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY WERE AS FCLLOWS:

1146	LOWER BOUND	UPPER BCUND	
	00-00-11-0	20.000.00	
30 04:463.0	20-001163-0	20-05-100	
0.862590 02	0.231780-02	0.231780-02	
0.665380 02	0.403550-02	0.463550-02	
0.906180 02	0-203160-01	0.203160-01	
C. 53C57C 02	0.226330-01	0.226335-01	
6.553770 62	0.352140-01	0.352146-01	
0.576560 02	0.910310-01	0.910310-01	
C.99435C 02	0.137910 00	0.137910 00	
C.162210 03	0.216700 00	0.216700 00	
0.104450 03	0.375830 00	0.37580C 00	
0.106770 03	0.010850 00	0.510850 00	
0-107+70 03	0.010850 00	0.510850 00	
0.106170 63	0. 764810 00	0.704810 00	
0.108865 03	0.754810 00	0.764810 00	
C.105560 C3	0.783910 00	0.780910 00	
0.110260 63	0.903240 00	0.408240 00	
0-110550 03	0.906240 00	0.90824C 00	
0.111650 63	0.908240 00	0.908240 00	
0.112350 63	0.908240 00	0.908240 00	
0-113040 63	0.100000 01	10 00000 01	

0.200000 01)

THE BCUNCS BERE CETERMINED USING THE UNION-CLUSTER PROCEDURE.

CNLY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS WERE CONSIDERED.

SYSTEMATIC SAPPLING BAS USED.

THE LPFER AND LCREE BCUNCS ON THE DISCRETE SUBNETWORK DURATION DISTRIBUTION ARE EQUAL.
THUS AN ESTIMATE OF THE DISCRETE DURATION DISTRIBUTION FOR SUBNETWORK 3 IS

NO																			
-	-02	-05	-05	0	-01	10-	-01	00		00					ô	00			
DISTRIBUT	-231780-	231780-	-03550+	2031605	226330-0	352140-	-015	137910	216700	375500	510850	510350	0180	080	780910	904240	240	240	938240
ISTA	231	153.		.203	.226	352	15016	137	.210	375	.510	.510	.76	.7640	.780	505	93824	.90824	906
٥	•	•	•	•	•	•	0	0	•	0	۰	•	•	•	•	•	•	•	0
	23	05	05	20	05	05	CS	05	C	03	3	03	03	C3	63	63	63	60	63
	290	350	380	190	015	170	299	350	2210	490	27.0	910	2	900	260	266	035	950	2350
	628	6025	885380	361805	5065	5537	976560	56535	0	*0	1067	1074	10817	1066	1095	.110	1105	=	=
=		:	:		;					;	;	;	:			:	:		

THIS IS THE CUTPLT FROM THE SYNTHESIS PROGRAM: SYNTH

THE CHARENT PROJECT SCHEOULE IS THE ONE MOST RECENTLY LISTED BY THE DETERMINISTIC SCHEOULE RESOLUTION PROGRAM. THE FRCJECT'S CERRESPENDING APPROXIMATE COMPLETION TIME DISTRIBUTION IS DETERMINED BELOW:

FCR THIS FACELEM

THE ALWEER CF C.D.F. SUBDIVISIONS USED THROUGADUT SYNTHESIS IS 49.
THE TAGET THRE USED BY THE DEFERMINISTIC SCHEDULER WAS 112.07396
THE WEAR CF THE PRCJECT'S APPROXIMENTE COMPLETION TIME DISTRIBUTION WILL BE COMPARED TO THE SPECIFIED PROJECT DEACLINE TIME OF 125.00000

THE INSTRUCTIONS TO BE PERFORMED ARE

SERIES=0
INSTRICTION NO. SLBNETROFK PARALLEL=! SUBNETWORKS TO BE SYNTHESIZED

m ~

APPROXIMATE (01STR 18UT 10N	DETERNINED	OSING	SUBNETWORK ANALYSI	ANALYSI
*	F(x)				
10.0000	0.0				
12.7843	0.05000				
15.2036	0.15000				
16.6132	0.25000				
17.8740	0.35000				
18.5984	00054.0				
22.7115	0.55000				
22.7115	0.0059-0				
25.2538	0.15000				
25.2538	0.85000				
30.3384	0.95000				
32.6607	1.00000				
*	F(x)				
83.9793	0.00232				
£6.2587	0.00232				
88.5382	0.00464				
\$0. E176	0.02032				
1250.65	0.02263				
65.3765	0.03521				
67.6560	0.09103				
99.9354	0.13791				
102-2148	0.21670				
104.4543	0.37580				
106.7737	0.51085				
167.4702	0.51085				
108.1667	0.76481				
108-8632	0.76481				
109.5597	0.78091				
110.2562	0.50824				
110.5527	0.90824				
111.6492	0.90824				
112.3457	0.90824				
113.0422	1.00000				

LENETBOR

THE DISTRIBUTION FOR THE SYNTHESIZED NETWORK IS

F(T)	0.0	0.0		.0001	00.	.0003	.0003	.0007	000	23.	0	.002	.000	00.	500.	.10.	0.01533	110.	.0454	.030	65.0.	. 6551	0.07352	. 0890	.1018	.1.3	. 1842	.2144	. 2964	0.31071	.3903	.4220		.535	519.	0.68729		-		. 83¢		0.50152	0.51587	•	0.50637	0.57506	0.57987	1.555.0	50	.0000
•		55.63541	0.0	7.1		5.2	;	101.35986		163.52001	104.56006	105.64016	106.70023	7.7	108.82036	5.650	*5.0	2.30	30.6	120	5.160	6.240	7.366	135.8	9.421	C.461		2.6613	3.661	4.7215	5.781		110	6.5c1	0.021	1.061	5.1	133.26216	:		•			139.56255	140.62262	141.68265	142.74277	143.80264	90	

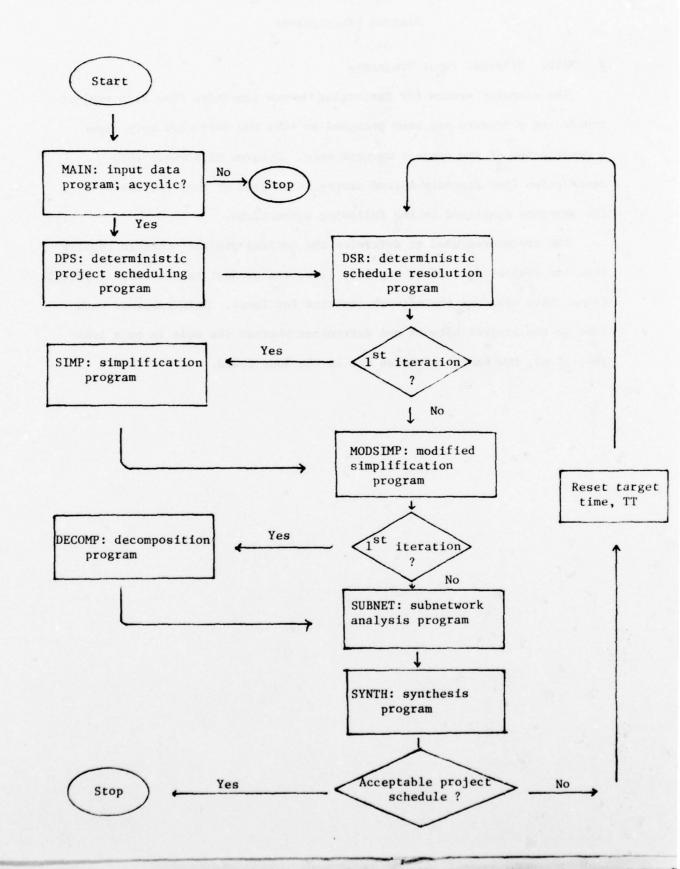
THE MEAN OF THE SYNTHESIZED DISTRIBUTION IS 128.42171

THE CIFFERENCE BETWEEN 128.42171 AND THE PROJECT DEADLINE IS PENCE. THE NEW TAMGET TIME IS 109.00782

2.74 PERCENT OF THE PROJECT DEADLINE.

THIS CCHPLETES ITERATICA

Appendix D. Flowchart of the Computer System



Appendix E

Program Descriptions

1. MAIN: Original Input Program

The computer system for performing the new iterative five step project scheduling procedure has been designed so that the user need only input a description of the project network once. Program MAIN reads this description (See Appendix A) and stores it on disk or tape for use by the programs described in the following subsections.

The procedures used to determine the optimal project schedule require that the project network be acyclic; i.e., the project has no loops (cycles). Hence, MAIN searches the network inputted for loops. MAIN examines each node in the project network and determines whether the node is on a loop and, if so, how many activities are in the loop found.

2. DPS: Deterministic Project Scheduling

The problem of finding a minimum cost project schedule which completes the project by TARGET TIME when each activity's duration is exactly its mean duration can be formulated as a linear programming problem. However, due to the large number of variables and constraints involved, a straightforward linear programming solution would be impractical. Instead the dual of this linear programming problem is considered, further reformulated, and then solved using the very efficient network-flow algorithm described in Dunn and Sielken (1977). This network-flow algorithm is a generalization of D. R. Fulkerson's algorithm (1961) for solving similar problems with linear activity cost functions. The generalized network-flow algorithm iteratively generates the minimum cost project schedule for every feasible deterministic completion deadline. The corresponding deterministic project cost curve is a convex piecewise linear function of TARGET TIME and a valuable description of the relationship between a project's cost and its deadline. The optimal activity mean durations are linear functions of the TARGET TIME on each linear piece of the project cost curve.

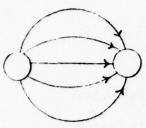
3. DSR: Deterministic Schedule Resolution

Since the deterministic project scheduler iteratively generates the optimal activity mean durations for all feasible completion deadlines, Step 1 is essentially only performed once. When the general iterative algorithm returns to Step 1 with a new TARGET TIME, finding the optimal activity mean durations is essentially a simple table look-up procedure. For a more complete detailed discussion of Deterministic Scheduling see Dunn and Sielken (1977).

4. SIMP: Simplification

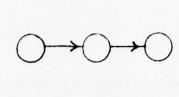
Five configurations of activities for which a single equivalent activity and duration distribution are readily available are depicted in Figure E.l. The equivalent single activity duration distributions for the parallel, series, and Wheatstone Bridge configurations were originally identified by Hartley and Wortham (1966) and for the Double Wheatstone Bridge and Criss-Cross configurations by Ringer (1969).

Simplification is an iterative procedure as illustrated in Figure E.2. In the special case where Simplification reduces the project network down to just one activity as in Figure E.2, the project completion time distribution is directly determined so that Steps 3 and 4 are skipped. Although a reduction to one activity is a very special case, reductions of over 50% are quite common.

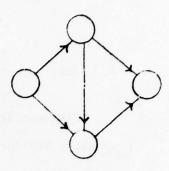


Activities in

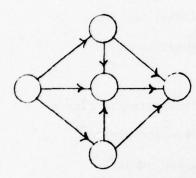
Parallel



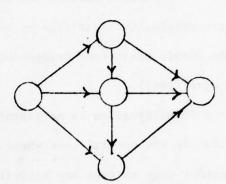
Two Activities in Series

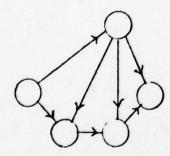


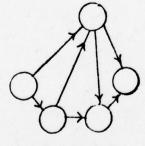
Wheatstone Bridge



Double Wheatstone Bridges



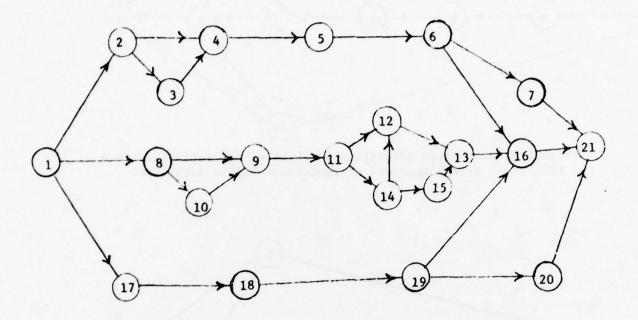




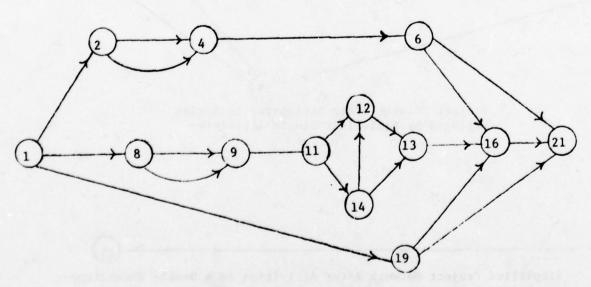
Criss - Crosses

Figure E.1 Activity Configurations Which Can be Readily Replaced by a Single Equivalent Activity

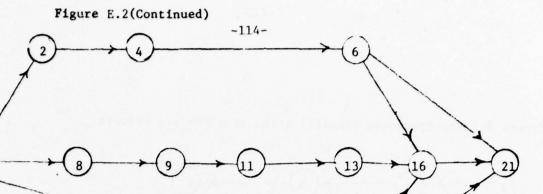
Figure E.2 The Iterative Simplification of a Project Network



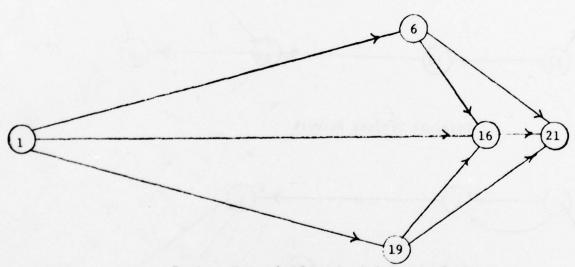
Original Project Network



Project Network After Activities in Series Replaced by Equivalent Single Activities



Project Network After Activities in Parallel and Activities in a Wheatstone Bridge Replaced by Equivalent Single Activities



Project Network After Activities in Series Replaced by Equivalent Single Activities

Simplified Project Network After Activities in a Double Wheatstone Bridge Replaced by an Equivalent Single Activity

5. MODSIMP: Subsequent Simplification

The structure of the simplified network is identified by SIMP.

MODSIMP determines the distribution of the activity durations in
the simplified network.

6. SUBNET: Subnetwork Analysis

6.1 Introduction

The objective of Subnetwork Analysis is to determine each subnetwork's duration distribution.

At the end of Step 2 each activity in the subnetwork has a specified duration distribution. This distribution is now approximated by a two-point discrete distribution. In particular, an activity, say A, is now conceptualized as having two possible duration times, say ℓ_A for a lower duration and u_A for an upper duration. The probability that the activity duration is ℓ_A is assumed to be P_A , and correspondingly the probability that the activity duration is u_A is assumed to be $Q_A = 1 - P_A$. The values of ℓ_A , u_A , and P_A are chosen so that the mean, variance, and third moment of the discrete distribution are the same as the mean, variance, and third moment of activity A's specified duration distribution.

Let n be the number of activities in the subnetwork. Let $v = 1, 2, ..., 2^n$ index the 2^n possible configurations of activity durations when each activity is either at its upper duration or at its lower duration. Let

p_v = probability of the v-th activity
duration configuration

$$= \prod_{i=1}^{n} [P_i(1 - \delta_{v,i}) + Q_i \delta_{v,i}]$$
 (6.1)

where

- $\delta_{v,i}$ = 1 if the duration for the i-th activity is u_i in the v-th activity duration configuration
 - = 0 if the duration for the i-th activity is l_i
 in the v-th activity duration configuration. (6.2)

Then the subnetwork duration distribution when each activity has its two-point discrete distribution is

$$F(t) = \sum_{v=1}^{2^{n}} p_{v} I_{t}(t_{v})$$
 (6.3)

where

 t_v = the subnetwork duration when the activity durations are in the v-th configuration (6.4)

and

$$I_t(t_v) = 1$$
 if $t_v \le t$,
= 0 if $t_v > t$. (6.5)

The discrete distribution function F is an approximation to the subnetwork's exact duration distribution.

The goal of Subnetwork Analysis is to determine F.

Since the number, n, of activities in the subnetwork may be fairly large, the complete enumeration of the 2ⁿ discrete subnetwork durations may sometimes be impractical. When this happens, the discrete subnetwork duration distribution F must be approximated. The approximation of F will be based on the activities which are mostly likely to influence the subnetwork duration. The identification of these important activities and their interrelationships is discussed in the next subsection which is a review of the procedures originating in Sielken,

Ringer, Hartley, and Arseven (1974) and Sielken, Hartley, and Spoeri (1976).

Each subnetwork is assumed to be an acyclic network with one source, one sink, and no cut vertices.

6.2 Formation of Clusters

The mean duration for activity A is

$$m_{A} = P_{A} \ell_{A} + Q_{A} u_{A} , \qquad (6.6)$$

and the standard deviation of activity A's duration is

$$s_{A} = [P_{A} \ell_{A}^{2} + Q_{A} u_{A}^{2} - m_{A}^{2}]^{\frac{1}{2}}. \qquad (6.7)$$

When each activity duration takes on a fixed (nonrandom) value, the subnetwork's duration is the duration of the longest path through the subnetwork where the "length" of an activity is its duration. For example, consider the subnetwork described in Table E.1 and displayed in Figure E.3. When each activity duration is its mean duration, then the subnetwork's duration is 32, corresponding to the path consisting of activities 2, 7, and 9.

<u>Definition 1</u>: A <u>critical</u> activity is an activity on the longest path when all the subnetwork's activity durations are set to their means.

Thus in the example the critical activities are 2, 7, and 9.

The search for the activities which are most likely to influence the subnetwork duration begins with the critical activities. Each critical activity initiates a separate set of activities called a

TABLE E.1

Activity Durations for the Subnetwork in Figure 2

Activity	^L A	u _A	PA	^m A	SA
1	0.00	2.00	.5	1	1
2	8.00	10.50	.2	10	1
3	9.55	15.67	.6	12	3
4	1.50	4.00	.8	2	1
5	3.35	5.52	.7	4	1
6	4.00	6.00	.5	5	1
7	8.73	16.90	.6	12	4
8	12.00	14.50	.2	14	1
9	5.00	15.00	.5	10	5

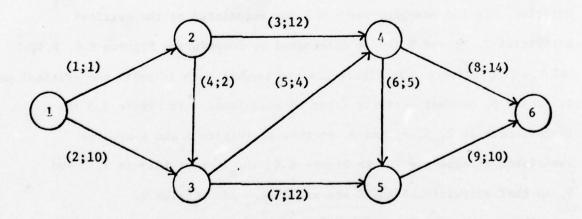


Figure E.3
Subnetwork with activities labeled (activity number; mean duration).

"cluster". Initially there are several clusters. In the example the initial clusters are

$$C_1 = \{2\}, C_2 = \{7\}, \text{ and } C_3 = \{9\}.$$
 (6.8)

Some of the non-critical activities may influence the subnetwork's duration when not all of the activity durations are at their mean values.

Definition 2: An associate of a critical activity A is a non-critical activity which is on the longest path when all activity durations are set to their mean except for the critical activity A which has its duration reduced from m_A to $max(m_A - \lambda s_A)$, 0) where λ is a nonnegative parameter.

Thus the associates of a critical activity A are those activities whose effect on the subnetwork's duration are related to activity A's duration. In the example, for λ = 1 the associates of the critical activities 2, 7, and 9 can be determined by considering Figures E.4, E.5, and E.6 respectively. In Figure E.4 the longest path is still the critical path 2, 7, and 9, so that activity 2 has no associates. In Figure E.5 the longest path is 2, 5, 6, and 9, so that activities 5 and 6 are the associates of activity 7. In Figure E.6, the longest path is 2, 5 and 8, so that activities 5 and 8 are associates of activity 9.

The associates of each critical activity are determined and added to the cluster containing that critical activity. Thus, in the example the clusters are expanded to

$$C_1 = \{2\}, C_2 = \{7, 5, 6\}, \text{ and } C_3 = \{9, 5, 8\}.$$
 (6.9)

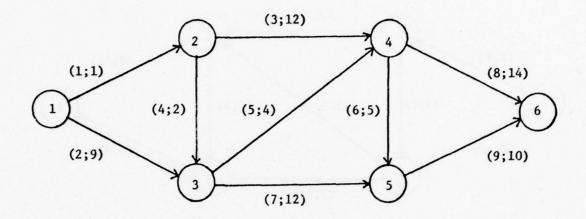


Figure E.4

Subnetwork for determining the associates of Activity 2 when λ = 1.

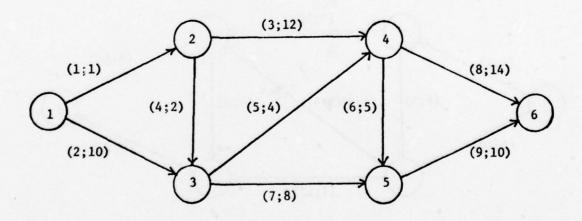


Figure E.5

Subnetwork for determining the associates of Activity 7 when λ = 1.

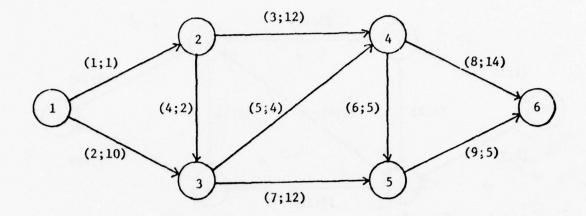


Figure E.6

Subnetwork for determining the associates of Activity 9 when λ = 1.

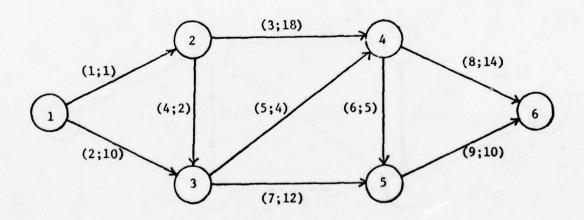


Figure E.7

Subnetwork for determining the eliminants of Activity 3 when θ = 2.

The idea underlying the clusters is that they should be sets of activities whose effects on the subnetwork's duration are interrelated. Thus, if two clusters contain any activities in common, the activities in these two clusters all have an interrelated effect on the subnetwork's duration, so the two clusters are combined into one cluster. In the example clusters C_2 and C_3 both contain activity 5, so they are combined. The resulting clusters are

$$C_1 = \{2\}$$
 and $C_2 = \{5, 6, 7, 8, 9\}$. (6.10)

A non-critical activity may also influence the subnetwork's duration if its duration exceeds its mean.

Definition 3: An eliminant of a non-critical activity A is a critical activity which is not on the longest path when all activity durations are set to their means except for activity A which has its duration increased from m_A to m_A + θs_A where θ is a nonnegative parameter.

For instance, if θ = 2, the eliminants of the non-critical activity 3 in the example can be determined from Figure E.7. There the longest path is 1, 3, 6, and 9, so that the eliminants of activity 3 are the critical activities 2 and 7. In the example, when θ = 2, none of the other non-critical activities (1, 4, 5, 6, and 8) have any eliminants. For a specified value of θ the eliminants of every non-critical activity are determined. If a non-critical activity A has eliminants, then the effect of A's eliminants on the subnetwork duration is related to A's duration, so A is added to every cluster containing at least one

of its eliminants. Thus in the example the clusters become

$$C_1 = \{2, 3\} \text{ and } C_2 = \{3, 5, 6, 7, 8, 9\}$$
. (6.11)

After the clusters have been expanded on the basis of eliminants, any two clusters containing common elements are combined. Therefore in the example, C_1 and C_2 are combined to form a single cluster

$$C_1 = \{2, 3, 5, 6, 7, 8, 9\}$$
 (6.12)

In general, after the determination of associates and eliminants for specified values of λ and θ and the subsequent combining of clusters, there may still be more than one cluster and some of the non-critical activities may not be in any cluster. Usually the larger the values of λ and θ the greater the number of activities in the clusters and the smaller the number of clusters. The clusters that remain represent sets of activities such that the effects on the subnetwork's duration of the activity durations for the activities within a set are all interrelated. Activities in different clusters have roughly independent effects on the subnetwork's duration. Activities not in any cluster have essentially no effect on the subnetwork's duration.

The consideration of critical activities, associates, eliminants, and the formation of clusters of related activities is obviously only one way of identifying the activities which have an important effect on the subnetwork's duration and their interrelationships. However, this particular procedure does have the following desirable properties:

- Property 1: If $\lambda_2 > \lambda_1$, then any activity which would be an associate of a critical activity A when $\lambda = \lambda_1$ would also be an associate of A when $\lambda = \lambda_2$.
- Property 2: If $\theta_2 > \theta_1$, then any critical activity which would be an eliminant of a non-critical activity A when $\theta = \theta_1$ would also be an eliminant of A when $\theta = \theta_2$.
- Property 3: For any fixed value of λ , the set of activities in the union of the clusters is monotically nondecreasing as $\theta \to \infty$.
- Property 4: The number of clusters is nonincreasing as $\theta + \infty$.
- Property 5: If $s_A > 0$ for a non-critical activity A, then there exists $\theta_A < \infty$ such that A will have some eliminants for any $\theta \geq \theta_A$.
- Property 6: If $s_A > 0$ for every non-critical activity A and

 $\theta* = \max\{\theta_A; A \text{ non-critical}\}$,

then for $\theta \ge \theta \star$ all activities will be in one cluster.

Most of these properties are fairly straightforward; however, Property 6 requires some special justification. This justification is based on the following definition and theorem which is proven in Sielken, Ringer, Hartley, and Arseven (1974).

<u>Definition 4</u>: In any acyclic network a <u>bridge</u> over any two consecutive arcs A_1 and A_2 is any arc A_3 such that all paths from the source to the sink passing through A_3 do not pass through either A_1 or A_2 .

Theorem 1: In any acyclic network with no cut vertices there is at least one bridge for any pair of consecutive arcs.

Property 5 implies that all activities will belong to some cluster if $\theta \geq \theta^*$. Now consider any two consecutive activities A_1 and A_2 on the critical path. Theorem 1 implies that there is a bridge over A_1 and A_2 , say A_3 . Since the critical path passes through A_1 and A_2 , A_3 cannot be on the critical path. Therefore, if $\theta \geq \theta^* \geq \theta_{A_3}$, A_1 and A_2 will be eliminants of A_3 and hence will be in the same cluster as A_3 . Thus, since each cluster contains at least one original critical activity and any two consecutive critical path activities belong to the same cluster when $\theta \geq \theta^*$, there is only one cluster when $\theta \geq \theta^*$ and Property 6 is established.

- 6.3 Bounding the Discrete Subnetwork Duration Distribution F
- 6.3.1 Upper Bounds on F

Suppose that the cluster formation procedure described in subsection 6.2 has been carried out on a subnetwork for some specified values of θ and λ and yielded K clusters. For each cluster C so determined, let n_c be the number of activities in the cluster and let $v=1,\ldots,2^{n_c}$ index the 2^{n_c} configurations of activity durations corresponding to

- (a) the duration for each activity A not in C being equal to its lower point $\ell_{\rm A}$, and
- (b) the durations for the activities in C being at each of the $2^{n_{\rm C}}$ possible combinations of their upper and lower points.

Then define

$$F^{+}(C; t) = \sum_{v=1}^{2^{n_{C}}} p_{v} I_{t}(t_{v})$$
 (6.13)

where p_v , t_v , and $I_t(t_v)$ are defined in (6.1), (6.4), and (6.5) respectively. The distribution function $F^+(C; t)$ is an upper bound on F. This can be shown by considering the following:

Theorem 2: For any cluster C, any t, and any activity Λ not in C,

$$F^{+}(C \cup \{A\}; t) \leq F^{+}(C; t)$$
.

(For the proof of this theorem, see Sielken, Hartley, and Spoeri (1975).) A straightforward application of Theorem 2 yields

Theorem 3: For any two clusters C_1 and C_2 and any t,

$$F^+(C_1 \cup C_2; t) \leq \min\{F^+(C_1; t), F^+(C_2; t)\}$$
.

If C* represents the set (cluster) of all activities in the subnetwork, then

$$F(t) = F^{+}(C^{*}; t)$$
 (6.14)

Since C is a subset of C*, either Theorem 2 or Theorem 3 implies

$$F^{+}(C; t) \geq F(t)$$
 (6.15)

for any cluster C.

Theorems 2 and 3 can also be used to define some tighter upper bounds on the subnetwork's duration distribution than $F^+(C; t)$. Historically, two different improved bounds have been employed, and both have been incorporated into the current subnetwork analysis

procedure. They are

$$F_{1}^{+}(t; \theta, \lambda) = F^{+}(\bigcup_{i=1}^{K} C_{i}; t)$$
 (6.16)

and

$$F_2^+(t; \theta, \lambda) = \min_{1 \le i \le K} F^+(C_i; t)$$
 (6.17)

Let $F^+(t; \theta, \lambda)$ denote either $F_1^+(t; \theta, \lambda)$ or $F_2^+(t; \theta, \lambda)$. Then, since Property 2 of the cluster formation procedure implies that as θ increases the clusters expand or are combined, Theorems 2 and 3 imply that $F^+(t; \theta, \lambda)$ is a nonincreasing function of θ for every t and any λ . Property 6 and (6.14) imply that for $\theta \geq \theta \star$

$$F^{+}(t; \theta, \lambda) = F(t) \tag{6.18}$$

for every t and any λ . Also (6.14) along with the definitions (6.16) and (6.17) imply

$$F^+(t; \theta, \lambda) > F(t)$$
 (6.19)

for all t, θ , and λ . These results are summarized in Theorem 4.

- Theorem 4: (a) $F^+(t; \theta, \lambda)$ is a nonincreasing function of θ for every t and any λ ;
 - (b) there exists a finite value θ^* such that $\theta \geq \theta^*$ implies $F^+(t; \theta, \lambda) = F(t)$ for every t and λ ; and
 - (c) for any θ , λ , and t

$$F^+(t; \theta, \lambda) \geq F(t)$$
.

6.3.2 Lower Bounds on F

Let n_c denote the number of activities in cluster C, and let $v=1,\ldots,2^c$ index the 2^c configuration of activity durations corresponding to

- (a) the duration for each activity A not in the cluster being equal to its upper point u_{Λ} , and
- (b) the durations for activities in the cluster being at each of the 2 c possible combinations of the upper and lower points.

Then define

$$F^{-}(C; t) = \sum_{v=1}^{n} p_{v} I_{t}(t_{v})$$
 (6.20)

where p_v , t_v , and $I_t(t_v)$ are as previously defined. Also define

$$F_{1}(t; \theta, \lambda) = F(UC_{1}; t)$$
 (6.21)

and

$$F_{2}(t; \theta, \lambda) = \max_{1 \le i \le K} F(C_{i}; t)$$
 (6.22)

Using an argument completely analagous to that used to prove Theorem 4, Sielken, Hartley, and Spoeri (1975) also proved

- Theorem 5: (a) $F(t; \theta, \lambda)$ is a nondecreasing function of θ for any fixed value of λ ;
 - (b) there exists a finite value $\theta *$ such that $\theta \ge \theta *$ implies

$$F^-(t; \theta, \lambda) = F(t)$$

for every t and any λ ; and

(c) for any θ , λ , and t

$$F(t; \theta, \lambda) \leq F(t)$$
.

(Again, $F^-(t; \theta, \lambda)$ is a generic term used to denote either $F_1^-(t; \theta, \lambda)$ or $F_2^-(t; \theta, \lambda)$.) Thus, $F^-(t; \theta, \lambda)$ is a valid lower bound on F.

6.3.3 The Tightness of the Bounds on F

That the F_1 -bounds are tighter than the F_2 -bounds can be seen as follows. The evaluation of $F_2^-(t;\,\theta,\,\lambda)$ involves the determination of $F_1^-(C_1;\,t)$ for each I whereas $F_1^-(t;\,\theta,\,\lambda) = F_1^-(U|C_1,\,t)$. Let I_1 be the length of the longest path when

- 1) the activities in $\mathbf{C}_{\mathbf{i}}$ are at a particular configuration of their upper and lower durations and
- 2) all activities not in $C_{\bf i}$ have their upper durations. Let $L_{\bf i}$ be the length of the longest path when
 - 1) the configuration of upper and lower durations for the activities in $C_{\bf i}$ is the same as in the determination of $L_{\bf i}$,
 - 2) the activities in U C $_j$ C $_i$ are at any combination of their upper and lower durations, and $_K$

$$F^{-}(UC_{j}; t) \ge F^{-}(C_{i}; t)$$
 (6.23)

and

$$F_1^-(t; \theta, \lambda) = F^-(\bigcup_{j=1}^K C_j; t) \ge \max_{1 \le i \le K} F^-(C_i; t) = F_2^-(t; \theta, \lambda)$$
. (6.24)

A similar argument can be used to show

$$F_1^+(t; \theta, \lambda) = F_2^+(UC_j; t) \leq \min_{1 \leq i \leq K} F_2^+(C_i; t) = F_2^+(t; \theta, \lambda)$$
 (6.25)

The extent of the differences between the two upper bounds and two lower bounds depends heavily on the structure of the particular subnetwork being analyzed and is a topic that should be considered in future empirical studies.

6.4 Using Sampling to Estimate the Upper and Lower Bounds on F

The only instance in which upper and lower bounds on F are computed rather than F itself is when it is computationally impractical to determine the longest path for each of the 2ⁿ activity duration configurations.

For given θ and λ , the evaluation of $F_1^+(t; \theta, \lambda)$ only requires the determination of the longest path for each of 2 activity configurations where n_U is the number of activities in the union of the clusters $C_0 = \bigcup_{j=1}^K C_j$; i.e.,

 $n_{\bigcup} = \sum_{j=1}^{K} n_{j} . \qquad (6.26)$

The evaluation of $F_1(t; \theta, \lambda)$ also entails only 2^{n_U} longest path determinations. Likewise, the evaluation of $F_2(t; \theta, \lambda)$ or $F_2(t; \theta, \lambda)$ only requires the determination of the longest path for each of

$$n_{s} = \sum_{i=1}^{K} 2^{n_{i}}$$
 (6.27)

activity configurations. Since 2 is always greater than or equal to

 n_s , $F_2^+(t; \theta, \lambda)$ and $F_2^-(t; \theta, \lambda)$ are the most economical bounds to compute in terms of the number of longest path determinations required. However, for any given θ and λ , $F_1^+(t; \theta, \lambda)$ and $F_1^-(t; \theta, \lambda)$ are tighter bounds than $F_2^+(t; \theta, \lambda)$ and $F_2^-(t; \theta, \lambda)$, respectively. Thus, in making the choice of which one of the two sets of bounds to compute, there is a trade-off between the accuracy of the bounds and the effort required to compute them.

Since the cluster formation procedure is such that the clusters expand or are pooled as θ increases, it may happen that for particular θ and λ , 2^{n_1} and 2^{n_2} for some i are both quite large even though θ is only moderately large. In this case it again becomes impractical to examine all the required activity configurations involved in determining either the F_1 -bounds or the F_2 -bounds. Consequently, if for the specified values of θ and λ , 2^{n_1} (or 2^{n_1} for some i, as the case may be) is excessively large, Subnetwork Analysis will compute estimates of the corresponding upper and lower bounds based on only a sample of the total number of possible configurations. The actual estimators of F^+ (C;t) or F^- (C;t) based on sample values $x_1 \leq x_2 \leq \ldots \leq x_m$ for cluster C is

$$G(t) = \frac{\sum_{i=1}^{m} P_{i}^{I} t^{(x_{i})}}{\sum_{i=1}^{m} P_{i}}$$
(6.28)

where p_i and $I_t(\cdot)$ are as in (6.1) and (6.5) respectively. The sample values are determined by finding the longest path for each member of a systematic sample of the possible activity duration configurations for

the cluster. However, if the cluster contains more than 31 activities, the sample values are determined by random sampling. The estimator, G(t), and the sampling procedures used in its evaluation are developed and discussed in Baker and Sielken (1978).

6.5 Estimating F by Extrapolating Between the Upper and Lower Bounds of F

Theorems 4 and 5 of subsection 6.3 imply that if $\theta_{i+1} \ge \theta_i$ and $\lambda_{i+1} \ge \lambda_i$ for all $i=1,\ldots,I$ then

$$F^{+}(t; \theta_{1}, \lambda_{1}) \geq F^{+}(t; \theta_{2}, \lambda_{2}) \geq ... \geq F^{+}(t; \theta_{1}, \lambda_{1}) \geq F(t) \geq$$

$$F^{-}(t; \theta_{1}, \lambda_{1}) \geq F^{-}(t; \theta_{1-1}, \lambda_{1-1}) \geq ... \geq F^{-}(t; \theta_{1}, \lambda_{1}) \qquad (6.29)$$

for all t. Thus, if $F^+(t; \theta, \lambda)$ and $F^-(t; \theta, \lambda)$ have been calculated for I pairs (θ_1, λ_1) i = 1, ..., I $(\theta_{1+1} \geq \theta_1, \lambda_{1+1} \geq \lambda_1)$, then F(t) may be estimated by extrapolating between $F^+(t; \theta_1, \lambda_1)$ and $F^-(t; \theta_1, \lambda_1)$. As currently written, Subnetwork Analysis calculates upper and lower bounds on the subnetwork's approximate duration distribution for a sequence of three (θ, λ) pairs, $(\theta, \lambda) = (1, 1)$, (2, 2), (3, 2). An extrapolation procedure is then used to obtain an estimate of F. The procedure that has been developed for this purpose is documented in Baker and Sielken (1978).

6.6 A Summary of the Subnetwork Analysis Procedure

The following is a step-by-step description of the subnetwork analysis procedure in summary form. Recall that the objective of

Subnetwork Analysis is to determine an "approximation", say F, to the subnetwork's duration distribution.

- (a) If n = 1, let F be the actual activity duration distribution for the one activity comprising the subnetwork, and stop.
 Otherwise, go to Step b.
- (b) Identify the two-point discrete distribution (ℓ_A , ι_A , P_A , Q_A) for every activity A in the subnetwork.
- (c) Ascertain the user's choice of
 - (1) NMAX, the maximum value of m for which all 2^m activity duration configurations are to be explicitly considered,
 - (2) the (θ, λ) pairs to be considered if not the standard pairs (1, 1), (2, 2), and (3, 2),
 - (3) whether the bounds on F are to be (F_1, F_1) or (F_2, F_2) if n > NMAX, and
 - (4) SAMSIZ, the sample size to be taken if, in the determination of bounds on F for some (θ, λ) pair, the number of activity configurations in the cluster being considered exceeds 2^{NMAX} .
- (d) If the number of activities in the subnetwork doesn't exceed NMAX, compute the subnetwork's discrete duration distribution, \hat{F} , explicitly, let $\hat{F} = F$, and stop. Otherwise, go to Step e.
- (e) Do Steps f i for every (θ, λ) pair. Then go to Step j.
- (f) Form the clusters corresponding to (θ, λ) . If the bounds are to be (F_1^-, F_1^+) , go to Step g. If the bounds are to be (F_2^-, F_2^+) , go to Step h.

- (g) Form the union of the clusters and determine n_U . If $n_U \leq \text{NMAX}$, evaluate the bounds (F_1^-, F_1^+) on the basis of all 2^{n_U} activity duration configurations. If $n_U > \text{NMAX}$, take a sample of size SAMSIZ from the 2^{n_U} activity duration configurations and form both F_1^- and F_1^+ on the basis of this single sample. Go to Step e.
- (h) Do the following for each cluster, C_i . Let n_i denote the number of activities in the cluster. If $n_i \leq \text{NMAX}$, evaluate $F^-(C_i; t)$ and $F^+(C_i; t)$ on the basis of all 2^{-1} activity duration configurations. If $n_i > \text{NMAX}$, take a sample of size SAMSIZ from the 2^{-1} activity duration configurations and form both $F^-(C_i; t)$ and $F^+(C_i; t)$ on the basis of this single sample.
- (i) Form F_2 and F_2 from the $F(C_i; t)$'s and $F(C_i; t)$'s respectively. Go to Step e.
- (j) Form \hat{F} by extrapolating the (F^-, F^+) bounds determined for the (θ, λ) pairs. Stop.

This process is repeated for every subnetwork in the simplified project network.

7. DECOMP: Decomposition

The objective of Deocmposition is to partition the simplified project network into the simplest possible subnetworks subject to the constaint that the subnetwork duration distributions can be easily combined to yield the project completion time distribution.

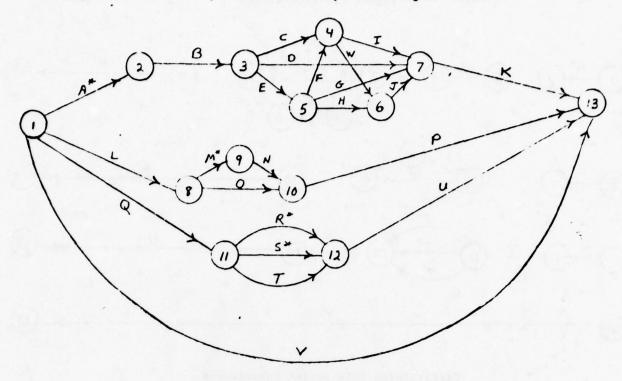
The simplified project network can be suitably partitioned using the following iterative procedure. First the simplified network is searched for subnetworks that begin at the beginning of the simplified network, end at the end of the simplified network, and are in parallel. Each parallel subnetwork is then subdivided into a sequence of smaller subnetworks that are in series. Each series subnetwork is then searched for parallel subnetworks.

The partitioning into parallel and series subnetworks continues until no subnetwork can be further partitioned. This iterative partitioning procedure is illustrated in Figure E.8.

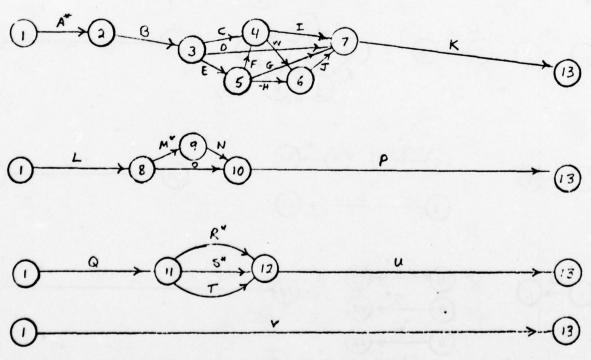
As in Simplification this partitioning of the simplified project network into subnetworks could have also included subnetwork configurations of the Wheatstone Bridge, Double Wheatstone Bridge, and Criss-Cross forms; however, the apparent frequency of these subnetwork configurations does not seem to justify the additional programming effort.

Since Steps 1 and 2 do not change the structure of the simplified project network and the partitioning of that network does not depend on the activity duration distributions, Decomposition is only done once and is really skipped when the general iterative algorithm returns to Step 3. Decomposition is documented in Sielken and Fisher (1976).

Figure E.8 The Becomposition of a Simplified Project Network*

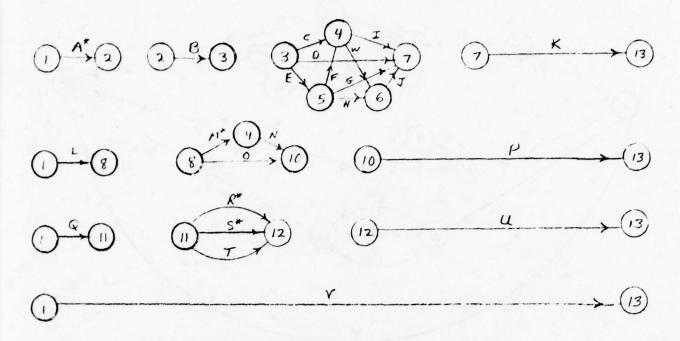


Simplified Project Network

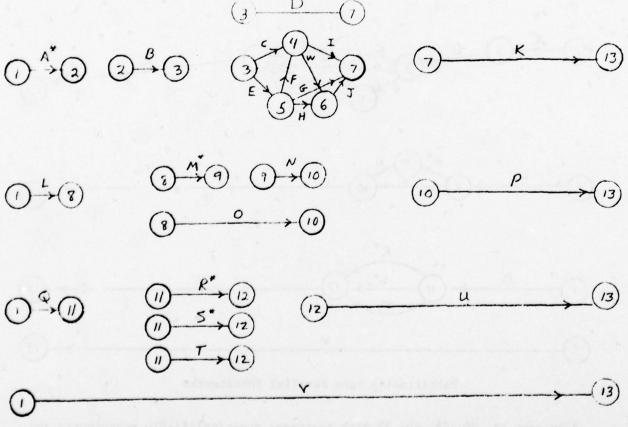


Partitioning into Parallel Subnetworks

The arcs A, M*, R*, and S* each represent non-simplifiable subnetworks and not single activities. Otherwise this "Simplified Project Network" could be further simplified.



Partitioning into Series Subnetworks



Final Partitioned Simplified Project Network

8. SYNTH: Synthesis

The project's approximate completion time distribution can be determined by combining the approximate subnetwork duration distributions. When the project network is decomposed in Step 3, the result is a network of subnetworks with any two connected subnetworks being either in series or in parallel. Let SUB_1 , and SUB_2 be any two such subnetworks, and let the corresponding approximate subnetwork duration distributions be $\hat{\mathrm{F}}_1$ and $\hat{\mathrm{F}}_2$. If SUB_1 and SUB_2 are in series, then the approximate duration distribution for SUB_1 and SUB_2 combined is

$$\hat{F}(t) = \sum_{s \le t} \hat{F}_2(t - s) \hat{f}_1(s) = \sum_{s \le t} \hat{F}_1(t - s) \hat{f}_2(s)$$
 (8.1)

where \hat{f}_1 and \hat{f}_2 are the discrete probability density functions corresponding to \hat{F}_1 and \hat{F}_2 respectively. If SUB₁ and SUB₂ are in parallel, then the approximate duration distribution for SUB₁ and SUB₂ combined is

$$\hat{F}(t) = \hat{F}_1(t) \cdot \hat{F}_2(t)$$
. (8.2)

By repeatedly combining subnetworks that are connected either in series or in parallel, the project's approximate completion time distribution is obtained.

Once the project's approximate completion time distribution has been determined, the project's approximate mean completion time, \hat{T} , can be calculated and compared with the project deadline. If the

project manager feels that T is sufficiently close to the project dealine, say within 5%, then the project schedule just determined in Step 1 is considered "optimal". Otherwise, a new project schedule must be determined by returning to Step 1 with a new TARGET TIME.

Step 1 only requires that the project would be completed by TARGET TIME if each activity's duration was exactly its mean duration. On the other hand, T takes into consideration the random nature of an activity's duration and hence will generally exceed TARGET TIME. The difficulty is in deciding how much less than the project deadline should TARGE TIME be in order that the corresponding T be sufficiently close to the project deadline. The algorithm iteratively updates its estimate of this TARGET TIME by

New TARGET TIME = Old TARGET TIME*(Project Deadline/T).

The initial TARGET TIME would usually be the project deadline but could be chosen somewhat less than the project deadline. A typical sequence of TARGET TIMEs and T's for a project deadline of 400 might be

TARGET TIME = 400, \hat{T} = 500, TARGET TIME = 320, \hat{T} = 360, TARGET TIME = 356, \hat{T} = 408.

Since the algorithm approximates the project's entire completion time distribution for each tentative schedule determined in Step 1, the quantity T in the above discussion could just as easily be a specified percentile of the project completion time distribution. For example, the project manager might wish a minimum cost project schedule such that the probability of the project being completed before the project deadline is .90. In this case, the new TARGET TIMEs would be determined with T being the 90th percentile of the project's approximate completion time distribution instead of the project's approximate mean completion time.

Of course, each time Steps 1-5 are performed a new T, approximate project completion time distribtion, and project cost are generated. This sequence of \hat{T} 's and project costs supplements the deterministic project cost curve in desribing the impact of the project deadline on the project cost.

Appendix F

Structure and Contents of the Temporary Data Sets

The project scheduling system requires eight temporary data sets which it uses to pass information from one program to the next. In the documentation, these data sets are called FILEO, FILE1, FILE2, ..., FILE7. The particular physical devices (tape, disk, etc.) which the file names represent are determined at execution time by the program's associated JCL-statements (Job Control Language statements) and by the user-defined unit numbers. The unit numbers corresponding to the eight data sets are defined symbolically in the FORTRAN code as FO, F1, F2, ..., F7, respectively. The particular values of the variables FO, ..., F7 are determined by DATAstatements which are inserted at the beginning of each program (see Appendix G for an example). Currently, F0 = 8, F1 = 9, F2 = 10, F3 = 11, F4 = 12, F5 = 13, F6 = 14, and F7 = 15. The particular form of the JCL-statements determine whether the unit numbers reference, tape, disk, or drum. The exact form of the JCL is highly dependent on the particular computer installation at which the system is being run. An example of the JCL used at Texas A&M is given in Appendix G.

The structure and contents of each of the data sets is outlined in the subsections that follow. In these subsections, variable names used in the input instructions (Appendix A) are not redefined. Any new variable names are defined when they are first used.

Written by: MAIN(A,B), DPS(C); i.e., parts A and B of FILEO are written by MAIN and part C is written by DPS

Modified by: none

Used by: DPS(A), DSR(A,C), SIMP(A,B), MODSIMP(A,B): i.e., DPS uses only part A of FILEO, etc.

- A. Record 1: $(4 \times MMAX + 5) \times 4$ bytes
 - 1. NACT; (Integer, I * 4)
 - 2. NODES: (I * 4)
 - 3. NSRCE; (I * 4)
 - 4. NSINK; (I * 4)
 - 5. LNODEN; (I * 4)
 - 6. S(MMAC,4); (two-dimensional array of I * 4 integers)

For $I = 1, \ldots, NACT$,

S(I,1) = the number of the I-th activity input, NA(I)

S(I,2) = the I-th activity's origin node, NODEO(I)

S(I,3) = the I-th activity's terminal node, NODET(I)

S(I,4) = the number of times and costs for the I-th activity, NCT(I)

For I > NACT,

S(I,1) = S(I,2) = S(I,3) = S(I,4) = 0

Currently, MMAX = maximum number of activites in the project network = 1000.

- B. Record 1: 4 bytes
 - 1. JMAT; (I * 4)

- C. Record 1: $(NBREAK + 1) \times 2$ bytes
 - 1. NBREAK; (Half-word integer, I * 2)

NBREAK = the number of times and costs needed to specify the entire project's time-cost curve

2. BREAK(NBREAK); (a one-dimensional array of I * 2 integers)

For $I = 1, \ldots, NBREAK$,

BREAK(I) = the I-th largest time in the entire project'stime-cost curve

Currently, NBREAK < 3000.

Written by: MAIN(A)

Modified by: none

Used by: DPS(A), DSR(A)

- A. Record 1: 8 bytes
 - 1. TEST1; (I * 2)
 - 2. TEST2; (I * 2)
 - 3. TEST3; (I * 2)

TEST3 = a computation option in DPS that is defined internally by MAIN = 0

4. STIME; (I * 2)

STIME = a parameter associated with TEST3 which is also defined internally by MAIN = 0

The remaining portion of FILE1 consists of one "block" of records for each original activity.

For $I = 1, \ldots, NACT$

Record $(I + 1) \cdot 1$: $(2 \times NCT(I)) \times 2$ bytes

- 1. TIME(1); (I * 2)
- 2. TIME(2); (I * 2)

NCT(I). TIME(NCT(I)); (I * 2)

NCT(I) + 1. COST(1); (I * 2)

NCT(1) + 2. COST(2); (1 * 2)

 $NCT(1) \times 2$. COST(NCT(1)); (1 * 2)

For J = 1, ..., NCT(I) - 1

Record (I + 1). (J + 1); 20 bytes

- 1. IDIST; (I * 4)
- 2. PARM1; (Real, R * 4)
- 3. PARM2; (R * 4)
- 4. PARM3; (R * 4)
- 5. PARM4; (R * 4)

IDIST, PARM1, PARM2, PARM3, and PARM4 define the distribution (and its parameters) of the I-th activity on the J-th segment of the time-cost curve

Written by: DSR(A)

Modified by: DSR(A)

Used by: MODSIMP(A)

A. One record for each original activity

For $I = 1, \ldots, NACT$,

Record I: 20 bytes

1. JDIST; (I * 4)

JDIST = the distributional type of the I-th activity

for the current TARGET TIME, TT

- 2. TPARM1; (R * 4)
- 3. TPARM2; (R * 4)
- 4. TPARM3; (R * 4)
- 5. TPARM4; (R * 4)

TPARM1, TPARM2, TPARM3, and TPARM4 are the I-th activity's distributional parameters for the current TARGET TIME, TT

Written by: MAIN(A), SIMP(B)

Modified by: DECOMP(C)

Used by: DECOMP(B), SUBNET(A,C)

A. Record 1: 44 bytes

- 1. IFDF; (I * 4)
- 2. NMAX; (I * 4)
- 3. IPOOL; (I * 4)
- 4. SAMSIZ; (I * 4)
- 5. THELAM; (I * 4)
- 6. THETA(1); (R * 4)
- 7. THETA(2); (R * 4)
- 8. THETA(3); (R * 4)
- 9. LAMBDA(1); (R * 4)
- 10. LAMBDA(2); (R * 4)
- 11. LAMBDA(3); (R * 4)

B. Record 1: $(4 \times MMAX + 5) \times 4$ bytes

This record is an exact duplicate of FILEO - part A except that the information refers to the simplified project network rather than the original project network. This record is <u>destroyed</u> by the execution of DECOMP and is replaced by part C.

C. Two records for each subnetwork identified by DECOMP

For K = 1, ..., number of subnetworks,

Record K.1: 4 bytes

1. M = the number of activities in the K-th subnetwork

Record K.2: $(3 \times M + 2) \times 4$ bytes

1. NSUB; (I * 4)

NSUB = the number of the K-th subnetwork

2. NMM; (I * 4)

NMM = the number of nodes in the K-th subnetwork

NET(3,M); (a two-dimensional array of I * 4 integers)

For I = 1, ..., M,

NET(2,I) = the terminal node of the I-th activity in the K-th subnetwork

NET(3,I) = the number assigned in the simplified project
 network to the K-th subnetwork's I-th activity

Written by: MAIN(A), MODSIMP(B)

Modified by: MODSIMP(B), SYNTH(A), READFIL(A)

Used by: DSR(A), SUBNET(B), SYNTH(A)

- A. Record 1: 12 bytes
 - 1. NCYC; (I * 4)

NCYC = the number of the current iteration

2. TT; (R * 4)

TT = TARGET TIME for the current iteration

3. NFLAG; (I * 4)

NFLAG = internally defined parameter used by DSR = 0 or 1

- B. Record 1: 4 bytes
 - 1. JNACT; (I * 4)

JNACT = the number of activities in the simplified
 project network

For $I = 1, \ldots, JNACT$,

Record I + 1: 96 bytes

1. CDFJ(12);(a one-dimenstional array of double word reals, R * 8)
 CDFJ(1), ..., CDFJ(12) = F⁻¹(.0), F⁻¹(.05), F⁻¹(.15),
 F⁻¹(.25), ..., F⁻¹(.95),
 F⁻¹(1.00) where F is the cumulative distribution function for the I-th activity in the
 simplified project network.

Written by: MAIN(A), DECOMP(B), SUBNET(C)

Modified by: SUBNET(C)

Used by: SYNTH(A,B,C)

- A. Record 1: 16 bytes
 - 1. IOPT; (I * 4)
 - 2. NT; (I * 4)
 - 3. PCT; (R * 4)
 - 4. PD; (R * 4)
- B. Record 1: 4 bytes
 - 1. NOINS; (I * 4)

NOINS ≈ the number of instructions generated by DECOMP for use in the synthesis process

Record 2; $(27 \times MAXINS) \times 4$ bytes

- INSNO(MAXINS); (a one-dimension array of I * 4 integers)
- 2. ISUBNT(MAXINS); (a one-dimension array of I * 4 integers)
- ISORP(MAXINS); (a one-dimension array of I * 4 integers)
- 4. JSUBNT (MAXINS, 24); (a two-dimensional array of I * 4 integers)

For $I = 1, \ldots, NOINS,$

INSNO(I) = the number of the I-th instruction

ISUBNT(I) ≈ the number of the subnetwork resulting from the I-th instruction

ISORP(I) = indicates whether the subnetworks involved
 in the I-th instruction are in parallel or series
 ISORP(I) = 0 implies the subnetworks are in
 series

ISORP(I) = 1 implies the subnetworks are in
 parallel

For J = 1, ..., 24,

JSUBNT(I,J) = the J-th subnetwork associated with the I-th
instruction

C. One record for each subnetwork identified by DECOMP

For $K = 1, \ldots,$ number of subnetworks,

Record K: $(2 \times JEDF + 2) \times 4$ bytes

1. ID; (I * 4)

ID = the number of the K-th subnetwork

2. JEDF; (I * 4)

JEDF = the number of subdivisions in the distribution function for the K-th subnetwork

3. XFD(JEDF,2); (a two-dimensional array of R * 4 reals)

For $J = 1, \ldots, JEDF$,

 ${\tt XFD}({\tt J,l})$ = the abscissa for the J-th tabled value of the cumulative distribution function for the K-th subnetwork

 ${\tt XFD}(J,2)$ = the ordinate for the J-th tabled value of the cumulative distribution function for the K-th subnetwork

Written by: DPS(A)

Modified by: none

Used by: DSR(A)

A. One record for <u>each</u> segment of the entire project's time-cost curve For $K = 1, \ldots, NBREAK$,

Record K: 4 × NACT bytes

1. XD(NACT, 2); (a two-dimensional array of I * 2 integers)

For $I = 1, \ldots, NACT$

XD(I,2) = the maximum duration of the I-th activity on the K-th function segment

Written by: SIMP(A)

Modified by: none

Used by: MODSIMP(A)

A. One record for \underline{each} activity simplification operation identified by SIMP

For $K = 1, \ldots,$ number of simplification operations,

Record K: 96 bytes

1. LIST(24); (a one-dimensional array of I * 4 integers)

For J = 1, ..., 24,

Table F.1

Contents of the Temporary Files at the Completion of Each Program

MAIN								
	A, 3	A	×	A	A	A	×	×
	A,B,C	А	×	A	A	A	A	×
DSR	A,B,C	А	A	А	А	A	A	×
SIMP	A,B,C	A	A	A,B	A	A	Ą	A
MODSIMP	A,B,C	A	×	A,B	A,B	A	A	A
DECOMP	A,B,C	A	×	A,C	A,B	A,B	Ą	A
SUBNET	A,B,C	A	×	A,C	A	A,B,C	A	A
SYNTH	A,B,C	A	×	A,C	A	A,B	А	A
second & successive iterations								
DSR	A,B,C	А	A	A, C	A	A, B	А	A
MODSIMP	A,B,C	Ą	X	A,C	A,B	A,B	A	¥
SUBNET	A,B,C	А	×	A,C	А	A,B,C	А	A
SYNTH	A,B,C	A	×	A,C	A	А,В	А	A

X means the file has no contents or that the information it does contain is never used in a subsequent step

Table F.2

€.

Files Referenced by Each Program

Temporary Files*

FILE7	×			×	×				×	×
FILE	×	×	×						×	×
FILES	×					×	×	×	×	×
FILE	×		×		×		×	×	×	×
FILE	×			×		×	×		×	×
61										
FILE	×		×		×				×	×
FILE	×	×	×						×	×
FILE(×	×	×	×	×				×	×
GRAM	IN	S	DSR	N.	SIMP	OMP	NET	ITH	FIL	FIL
PRO	MA	DF	DS	SI	MOD	DEC	SUB	SYN	SAVE	READ

* The temporary data sets referenced by each program in the computer system are indicated by an "X" in the appropriate column. Every data set referenced by a particular program should be defined for the job-step in which that program is executed.

Appendix G

Job Control Language for Texas A&M University Computer Facilities

1. Execution using FORTRAN source decks

A listing of the JCL used to generate the sample output of Appendix C is given below. During the first iteration each program in the sequence is compiled using the IBM FORTRAN G Compile and the load module which ultimately results is stored on the partitioned data set named USER.STAT.SIELKEN.PERT. Thus, on subsequent iterations the compile and link-edit steps are skipped.

//JOBLIB DD DSNAME=USER.STAT.SIELKEN.FERT.DISP=SHR //MAIN EXEC FORTGCL.REGION=256K //FORT.SYSIN DD *

INSERT FORTRAN COURCE DECK FCR MAIN HERE

```
//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(MAIN).DISP=SHR
//MAIN EXEC PGM=MAIN.REGION=320K,COND=(4,LT)
//FT06FU01 DD SYSDUT=A
//FT08F001
              DD UNIT=SYSDA.DSN=&&FILEO.DISP=(NEW.PASS).
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT09F001
              DD UNIT=SYSDA.DSN=&&FILE1.DISP=(NEW.PASS).
         SPACE = (32000, (100, 25)), D@B = (LRECL = 3152, BLKSIZE = 3156, RECFM = VBS)
//FT1UFOU1
              DD UNIT=SYSDA.DSN=&&FILE2.DISP=(NEW.PASS).
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT11F001
              DD UNIT=SYSDA.DSN=&&FILE3.DISP=(NEW.PASS).
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT12F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(NEW.PASS),
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT13F001
              DD UNIT=SYSDA.DSN=&&FILE5.DISP=(NEW.PASS).
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT14F001
              DD UNIT=SYSDA . DSN=&&FILE6 . DISF=(NEW . PASS) .
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
              DD UNIT=SYSDA, DSN=&&FILE7, DISP=(NEW, PASS),
11
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT35F001 DD *
```

INSERT DATA CARDS FOR MAIN HERE

//DPS EXEC FORTGCL.REGION=384K //FORT.SYSIN DD #

INSERT FORTRAN COURCE DECK FCR DPS HERE

```
//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(DPS).DISP=SHR
//DPS EXEC PGM=DPS.REGION=512K.COND=(4.LT)
//FTU0F001 DD SYSOUT=A
//FT0dF001 DD UNIT=SYSDA.DSN=&&FILE0.DISP=(OLD.PASS)
//FTJ9FU01 DD UNIT=SYSDA.DSN=&&FILE1.DISP=(OLD.PASS)
//FT14F001 DD UNIT=SYSDA.DSN=&&FILE6.DISP=(OLD.PASS)
//DSR EXEC FORTGCL.REGION=256K
//FORT.SYSIN DD *
```

INSERT FORTRAN COURCE DECK FOR DSR HERE

```
//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(DSR),DISP=SHR
//DSR EXEC PGM=DSR.REGION=256K.COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001 DD UNIT=SYSDA.DSN=&&FILEO.DISP=(OLD.PASS)
//FT09F001 DD UNIT=SYSDA.DSN=&&FILE1.DISP=(OLD.PASS)
//FT1UF001 DD UNIT=SYSDA.DSN=&&FILE2.DISP=(OLD.PASS)
//FT12F001 DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
//FT14F001 DD UNIT=SYSDA.DSN=&&FILE6.DISP=(OLD.PASS)
//FSIMP EXEC FORTGCL.REGION=320K
//FURT.SYSIN DD *
```

INSERT FORTRAN COURCE DECK FCR SIMP HERE

```
//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(SIMP).DISP=SHR
//SIMP EXEC PGM=SIMP.REGION=256K.COND=(4.LT)
//FT06F001 DD SYSOUT=A
//FT08F001 DD UNIT=SYSDA.DSN=&&FILEO.DISP=(OLD.PASS)
//FT11F001 DD UNIT=SYSDA.DSN=&&FILE3.DISP=(OLD.PASS)
//FT15FJ01 DD UNIT=SYSDA.DSN=&&FILE7.DISP=(OLD.PASS)
//MUDSIMP EXEC FORTGCL.REGION=256K
//FURT.SYSIN DD *
```

INSERT FORTRAN COURCE DECK FOR MCDSIMP HERE

```
//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(MODSIMP),DISP=SHR
//MODSIMP EXEC PGM=MODSIMP,REGION = 320K,COND=(4.LT)
//FT06F001 DD SYSOUT=A
//FT08F001 DD UNIT=SYSDA.DSN=&&FILEO.DISP=(OLD.PASS)
//FT10F001 DD UNIT=SYSDA.DSN=&&FILE2.DISP=(OLD.PASS)
//FT12F001 DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
//FT15F001 DD UNIT=SYSDA.DSN=&&FILE7.DISP=(OLD.PASS)
//DECOMP EXEC FORTGCL.REGION=256K
//FORT.SYSIN DD *
```

INSERT FORTRAN COURCE DECK FOR DECOMP HERE

```
//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.FERT(DECOMP).DISP=SHR

//DECOMP EXEC PGM=DECOMP.REGION=320K.CCND=(4.LT)

//FT06F001 DD SYSOUT=A

//FT11F001 DD UNIT=SYSDA.DSN=&&FILE3.DISP=(OLD.PASS)

//FT13F001 DD UNIT=SYSDA.DSN=&&FILE5.DISP=(OLD.PASS)

//SUBNET EXEC FORTGCL.REGION=256K

//FORT.SYSIN DD *
```

INSERT FORTRAN COURCE DECK FOR SUBNET HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(SUBNET).DISP=SHR //SUBNET EXEC PGM=SUBNET.REGION=448K,CCND=(4.LT) //FTJ6FJ01 DD SYSOUT=A

```
//FT11Fu01
//FT12F001
DD UNIT=SYSDA.DSN=&&FILE3.DISP=(OLD.PASS)
DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)

//FT13F001
DD UNIT=SYSDA.DSN=&&FILE5.DISP=(OLD.PASS)

//SYNTH EXEC FORTGCL.REGION=256K

//FORT.SYSIN DD *
```

INSERT FORTRAN COURCE DECK FCR SYNTH HERE

```
//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(SYNTH).DISP=SHR
//SYNTH EXEC PGM=SYNTH.REGION=256K.CCNC=(4.LT)
//FTOOFJOI DD SYSOUT=A
//FT12FJ01
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
              DD UNIT=SYSDA, DSN=&&FILE5, DISP=(OLD, PASS)
//FT13FJ01
//DSR EXEC PGM=DSR, REGION=256K, COND=(4,LT)
//FTUOFUU1 DD SYSOUT=A
              DD UNIT=SYSDA.DSN=&&FILEO.DISP=(OLD.PASS)
//FT08F001
//FT09F001
              DD UNIT=SYSDA, DSN=&&FILE1, DISP=(OLD, PASS)
              DD UNIT=SYSDA.DSN=&&FILE2.DISP=(OLD.PASS)
//FT10F001
//FT12F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
              DD UNIT=SYSDA.DSN=&&FILE6.DISP=(OLD.PASS)
//FT14F001
//MUDSIMP EXEC PGM=MODSIMP.REGION=320K.COND=(4.LT)
//FT06F001 DD SYSOUT=A
              DD UNIT=SYSDA, DSN=&&FILEC, DISF=(OLD, PASS)
//FT08F001
//FT10FJ01
              DD UNIT=SYSDA.DSN=&&FILE2.DISP=(OLD.PASS)
//FT12F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
              DD UNIT=SYSDA.DSN=&&FILE7.DISP=(OLD.PASS)
//FT15F001
//SUBNET EXEC PGM=SUBNET.REGION=448K.CEND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT11F001
              DD UNIT=SYSDA.DSN=&&FILE3.DISP=(OLD.PASS)
//FT12F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
//FT13F001
              DD UNIT=SYSDA. DSN=&&FILES.DISP=(OLD.PASS)
//SYNTH EXEC PGM=SYNTH.REGION=256K.CONC=(4,LT)
//FT06FJ01 DD SYSOUT=A
//FT12F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
//FT13F001
              DD UNIT=SYSDA, DSN=&&FILE5, DISP=(OLD, PASS)
/*END
```

2. Execution using load modules

If the load modules corresponding to the programs in the system have already been prepared and stored on the disk file USER.STAT.SIELKEN.PERT., the following JCL is appropriate.

```
//JOBLIB DD DSNAME=USER.STAT.SIELKEN.FERT.DISP=SHR
//MAIN EXEC PGM=MAIN.REGION=320K.COND=(4.LT)
//FTOOFOOI DD SYSOUT=A
//FT08F001
              DD UNIT=SYSDA, DSN=&&FILEO, DISP=(NEW, PASS),
         SPACE = (32000, (100,25)), DCB=(LRECL=3152, BLKSIZE=3156, RECFM=VBS)
//FT09F001
              DD UNIT=SYSDA.DSN=&&FILE1.DISP=(NEW.PASS),
11
         SPACE=(32000.(100.25)).DCB=(LRECL=3152.BLKSIZE=3156.RECFM=VBS)
//FT10F001
              DD UNIT=SYSDA, DSN=&&FILE2.DISP=(NEW.PASS).
11
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKS1ZE=3156,RECFM=VBS)
//FT11F001
              DD UNIT=SYSDA.DSN=&&FILE3.DISP=(NEW,PASS),
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT12F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(NEW.PASS).
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
11
//FT13F301
              DD UNIT=SYSDA, DSN=&&FILE5.DISF=(NEW.PASS).
11
         SPACE=(32000.(100.25)).DCB=(LRECL=3152.BLKSIZE=3156.RECFM=VBS)
//FT14F001
              DD UNIT=SYSDA.DSN=&&FILE6.DISP=(NEW.PASS).
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT15F001
              DD UNIT=SYSDA, DSN=&&FILE7, DISP=(NEW, PASS).
         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
11
//FT05F001 DD *
```

INSERT DATA CARDS FOR MAIN HERE

```
//DPS EXEC PGM=DPS.REGION=512K,COND=(4.LT)
//FTOOFOOL DD SYSOUT=A
              DD UNIT=SYSDA, DSN=&&FILEO, DISP=(OLD, PASS)
//FTOBFJO1
//FT09F001
              DD UNIT=SYSDA.DSN=&&FILE1.DISP=(OLD.PASS)
//FT14FUU1
              DD UNIT=SYSDA.DSN=&&FILE6.DISP=(OLD.PASS)
//DSR EXEC PGM=DSR, REGION=256K, COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001
              DD UNIT=SYSDA.DSN=&&FILEO.DISP=(OLD.PASS)
              DD UNIT=SYSDA.DSN=&&FILE1.DISP=(OLD.PASS)
//FTJ9FJJ1
//FTIUFO01
              DD UNIT=SYSDA.DSN=&&FILE2.DISP=(OLD.PASS)
              DD UNIT=SYSDA.DSN=&&FILE4.DISF=(OLD.PASS)
//FT12F001
              DD UNIT=SYSDA, DSN=&&FILE6.DISP=(OLD, PASS)
//FT14F001
//SIMP EXEC PGM=SIMP.REGION=256K.COND=(4.LT)
//FT06FUU1 DD SYSOUT=A
              DD UNIT=SYSDA.DSN=&&FILEO.DISP=(OLD.PASS)
//FTOSFOO1
              DD UNIT=SYSDA.DSN=&&FILE3.DISP=(OLD.PASS)
//FT11F001
              DD UNIT=SYSDA.DSN=&&FILE7.DISP=(OLD.PASS)
//FT15F001
//MODSIMP EXEC PGM=MODSIMP.REGION=320K.COND=(4.LT)
//FTU6FOOI DD SYSOUT=A
              DD UNIT=SYSDA.DSN=&&FILEO.DISP=(OLD.PASS)
//FTJ8F001
              DO UNIT=SYSDA.DSN=&&FILE2.DISP=(OLD.PASS)
//FT10F001
//FT12F001
              DD UNIT=SYSDA, DSN=&&FILE4.DISF=(OLD.PASS)
//FT15FJ01
              DD UNIT=SYSDA.DSN=&&FILE7.DISP=(OLD.PASS)
```

```
//DECOMP EXEC PGM=DECOMP.REGION=320K.COND=(4.LT)
//FTOOFOO1 DD SYSOUT=A
//FT11F001
              DD UNIT=SYSDA.DSN=&&FILE3.DISP=(OLD.PASS)
//FT13F001
              DD UNIT=SYSDA, DSN=&&FILE5.DISP=(OLD, PASS)
//SUBNET EXEC PGM=SUBNET.REGION=448K.CCND=(4.LT)
//FT06FU01 DD SYSOUT=A
              DD UNIT=SYSDA, DSN=&&FILE3, DISP=(OLD, PASS)
//FT11F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
//FT12F001
//FT13F301
              DD UNIT=SYSDA.DSN=&&FILE5.DISF=(OLD.PASS)
//SYNTH EXEC PGM=SYNTH.REGION=256K.CCND=(4.LT)
//FT06FJU1 DD SYSOUT=A
              DD UNIT=SYSDA.DSN=&&FILE4.DISF=(OLD.PASS)
//FT12F001
              DD UNIT=SYSDA.DSN=&&FILES.DISP=(OLD.PASS)
//FT13F001
//DSR EXEC PGM=DSR.REGION=256K,COND=(4.LT)
//FTU6FUU1 DD SYSOUT=A
//FTOBFOOL
              DD UNIT=SYSDA.DSN=&&FILEO.DISP=(OLD.PASS)
//FTJ9FJ01
              DD UNIT=SYSDA. DSN=&&FILE1.DISP=(OLD.PASS)
//FT1 JF 001
              DD UNIT=SYSDA.DSN=&&FILE2.DISP=(OLD.PASS)
//FT12F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
//FT14F001
              DD UNIT=SYSDA.DSN=66FILE6.DISP=(OLD.PASS)
//MODSIMP EXEC PGM=MODSIMP.REGION=320K.CONC=(4.LT)
//FT06F001 DD SYSOUT=A
              DD UNIT=SYSDA, DSN=&&FILEO.DISP=(OLD.PASS)
//FT08F301
              DD UNIT=SYSDA.DSN=&&FILE2.DISP=(OLD.PASS)
//FT10F001
//FT12FJ01
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
              DD UNIT=SYSDA.DSN=&&FILE7.DISP=(OLD.PASS)
//FT15F001
//SUBNET EXEC PGM=SUBNET.REGION=448K.CCND=(4.LT)
//FTO6FOOL DD SYSOUT=A
              DD UNIT=SYSDA.DSN=&&FILE3.DISP=(OLD.PASS)
//FT11F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
//FT12F001
//FT13F001
              DD UNIT=SYSDA.DSN=&&FILE5.D[SP=(OLD.PASS)
//SYNTH EXEC PGM=SYNTH.REGION=256K.COND=(4.LT)
//FT06F001 DD SYSOUT=A
//FT12F001
              DD UNIT=SYSDA.DSN=&&FILE4.DISP=(OLD.PASS)
//FT13F301
              DD UNIT=SYSDA.DSN=&&FILE5.DISP=(OLD.PASS)
/*END
```

Appendix H

Interrupting the Computer System

The number of iterations necessary to identify an "acceptable" project schedule using the new project scheduling procedure depends on many factors and usually cannot be determined in advance. Also the number of job steps allowed in the JCL-program for a given run varies widely from one computer installation to the next. (Currently, at Texas A&M the limitation is 10 job steps.) Thus, some provision must be made for interrupting the computer system prior to the identification of an acceptable project schedule and restarting it from the point of interruption.

One method is to let the "temporary" data sets defined in Appendix F correspond to user-supplied (permanent) storage devices rather than system-supplied (temporary) ones. This requires, however, that the user supply (and pay for)eight different permanent storage devices (or locations).

A second method which requires that the user supply only one permanent storage device involves the use of the programs SAVEFIL and READFIL which allow the user the capability of 1) interrupting the computer system at any of the starred points (*) indicated on Figure H.1 and 2) continuing the procedure from the point of interruption at some future time. SAVEFIL and READFIL are documented in subsections 1 and 2, respectively, of this appendix. Subsection 3 gives a listing of their associated JCL.

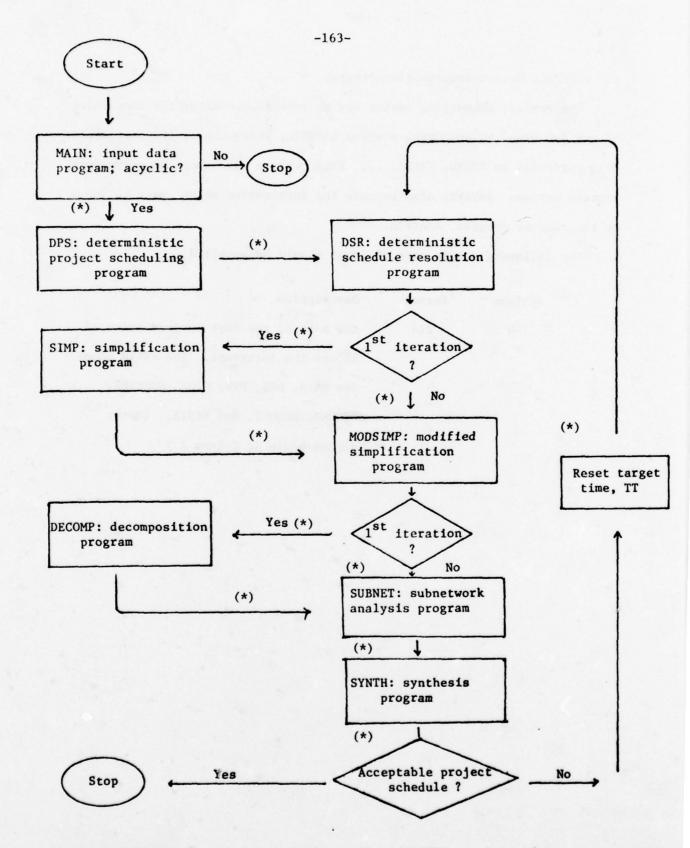


Figure H.1 Flowchart with (*) indicating where SAVEFIL may be used to interrupt the system.

1. SAVEFIL: System Interruption Program

The project scheduling system may be interrupted after the completion of any job step by executing program SAVEFIL, which copies the pertinent data currently on FILEO, FILEI, ..., FILE7 onto a user-supplied permanent storage device. SAVEFIL also outputs the information which must be input to the restart program, READFIL.

The following single card of input should be supplied:

Column	Format	Description
1-8	2A4	the name of the last program executed
		before the interrupt. The valid names
		are MAIN, DPS, DSR, SIMP, MODSIMP,
		DECOMP, SUBNET, and SYNTH. (Note:
		always begin in Column 1.)

2. READFIL: System Restart Program

When the project scheduling system has been interrupted by the use of SAVEFIL, READFIL must be used to restore the necessary information to the temporary data sets before the iterative procedure may resume. The following single card of input should be supplied:

Column	Format	Description
1-8	2A4	the name of the last program executed
		before the interrupt.
1-9	12	0 if the procedure was interrupted
		before the first execution of SYNTH
		1 otherwise

The remaining parameters should <u>only</u> be supplied if the procedure was interrupted immediately after the execution of SYNTH.

11-20	F10.5	The value of TT for the next iteration
		if different from the value specified
		by SYNTH.
21-25	15	1 if the values of any of the following
		are to be different from their
		current values in the next iteration:
		IOPT, PCT, PD
		0 otherwise
26-30	15	the new value for IOPT
31-40	F10.5	the new value for PCT
41-50	F10.5	the new value for PD

See Appendix A for the definition of TT, IOPT, PCT, and PD.

3. JCL for SAVEFIL and READFIL

The following exemplifies the JCL required by program SAVEFIL for the Texas A&M computer facilities. For this example, the user-supplied permanent storage device is named USER.STAT.SIELKEN.DATA.

//JOBLIB DD DSNAME=USER.STAT.SIELKEN.PERT.DISP=SHR //SAVEFIL EXEC FORTGCL.REGION=256K //FORT.SYSIN DD *

INSERT FORTRAN SOURCE DECK FOR SAVEFIL HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.FERT(SAVEFIL).DISP=SHR //SAVEFIL EXEC PGM=SAVEFIL . REGION=256K . COND=(4.LT) //FT04F001 DD DSN=USER.STAT.SIELKEN.DATA.DISP=SHR //FT06F001 DD SYSOUT=A //FTOBFOOI DD UNIT=SYSDA.DSN=&&FILEO.DISP=(OLD.DELETE) DD UNIT=SYSDA.DSN=&&FILE1.DISP=(OLD.DELETE) //FT09F001 DD UNIT=SYSDA. DSN=&&FILE2.DISP=(OLD. DELETE) //FT10F001 //FT11F001 DO UNIT=SYSDA, DSN=&&FILE3, DISP=(OLD, DELETE) //FT12F001 DD UNIT=SYSDA, DSN=&&FILE4, DISP=(OLD, DELETE) DD UNIT=SYSDA, DSN=6&FILE5.DISF=(OLD.DELETE) //FT13FJ01 //FT14F001 DD UNIT=SYSDA.DSN=&&FILE6.DISP=(OLD.DELETE) DD UNIT=SYSDA.DSN=&&FILE7.DISP=(OLD.DELETE) //FT15F301 //FT05F001 DD *

1

INSERT DATA CARD FOR SAVEFIL HERE

/#END

Similarly, the JCL for program READFIL is as follows:

//JOBLID DD DSNAME=USER.STAT.SIELKEN.FERT.DISP=SHR //READFIL EXEC FORTGCL.REGION=256K //FORT.SYSIN DD #

INSERT FORTRAN SOURCE DECK FCF READFIL HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(READFIL).DISP#SHR //READFIL EXEC PGM=READFIL . REGION=256K . CCND=(4.LT) DD DSN=USER.STAT.SIELKEN.CATA.CISP=SHR //FT04F001 //FTJoFOUL DD SYSOUT=A DD UNIT=SYSDA.DSN=&&FILEO.DISP=(NEW.PASS). //FTJ8F001 SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS) //FTJ9FJ01 DD UNIT=SYSDA.DSN=&&FILE1.DISP=(NEW,PASS). 11 SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS) //FT1UF001 DD UNIT=SYSDA.DSN=&&FILE2.DISP=(NEW.PASS). SPACE=(32000.(100.25)).DCB=(LRECL=3152.BLKSIZE=3156.RECFM=VBS) 11 //FT11F001 DD UNIT=SYSDA, DSN=&&FILE3, DISP=(NEW, PASS), SPACE=(32000,(100,25)),DCB≈(LRECL=3152,BLKSIZE=3156,RECFM=VBS) 11 //FT12F001 DD UNIT=SYSDA.DSN=&&FILE4.DISP=(NEW.PASS), SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS) 11 //FT13F001 DD UNIT=SYSDA.DSN=&&FILE5.DISP=(NEW.PASS). 11 5PACE=(32000,(100,25)),DCB≈(LRECL=3152,BLKSIZE=3156,RECFM=VBS) //FT14F001 DD UNIT=SYSDA.DSN=&&FILE6.DISP=(NEW,PASS). SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS) //FT15F001 DD UNIT=SYSDA, DSN=&&FILE7, DISP=(NEW, PASS). SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS) 11 //FT35F001 DD *

INSERT DATA CARD FOR READFIL HERE

//---- CONTINUE WITH THE JCL FOR THE NEXT PROGRAM IN THE SEQUENCE

Appendix I

Program Listings

1. MAIN

DO 1300 I=1.NACT

*

```
C
      MAIN PROGRAM
C
         THIS PROGRAM READS ALL USER SUPPLIED INPUT TO THE PROJECT
         SCHEDULING SYSTEM AND STORES THE INFORMATION ON (SCRATCH)
         DISK FILES DEFINED BY THE USER. THE INFORMATION IS RETIEVED
         AS NEEDED BY THE INDIVIDUAL PROGRAMS.
         SEE TECHNICAL REPORT #57 FOR SPECIFIC INPUT INSTRUCTIONS
                                                                                  10
         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET
                                                                                  11
         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL
                                                                                  12
C
                     PROJECT NETWORK
                                                                                  13
         LNMAX = THE LARGEST NGDE NUMBER ALLOWED IN THE PROJECT NETWORK
                                                                                  14
         NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR
                                                                                  15
                     EACH ACTIVITY
                                                                                  16
         CURRENTLY. MMAX=1000; LNMAX=9999; NCTMAX=6
                                                                                  17
C
                                                                                  18
      INTEGER#4 FILEO(4*MMAX+5).NA(MMAX)
                                                                                  19
      INTEGER+4 NGDEG(MMAX).NODET(MMAX).NCT(MMAX).IDIST(MMAX.NCTMAX-1)
                                                                                  20
      INTEGER+2 TIME(MMAX.NCTMAX).COST(MMAX.NCTMAX)
C
                                                                                  21
      REAL *4 PARMI (MMAX.NCTMAX-1).PARM2 (MMAX.NCTMAX-1)
                                                                                  22
C
      REAL *4 PARM3 (MMAX. NCTMAX-1). PARM4 (MMAX. NCTMAX-1)
                                                                                  23
C
      EGUIVALENCE (FILEO(6).NA(1)).(FILEO(MMAX+6).NODEO(1))
      EQUIVALENCE (FILEO(2*MMAX+6).NODET(1)).(FILEO(3*MMAX+6.NCT(1))
C
                                                                                  25
         IN SUBROUTINE LOOP, THE ADDITIONAL ARRAY DIMENSIONS ARE:
                                                                                  26
      DIMENSION A(LNMAX) . 8(LNMAX)
                                                                                  27
                                                                                  28
         THE VALUES OF THE VARIABLES FO - F7 SHOULD CORRESPOND TO UNIT
         NUMBERS FUR THE INSTALLATION'S TEMPORARY STORAGE DEVISES(CISKS)
C
                                                                                  30
         WHICH WILL BE REFERENCED BY THE PROJECT SCHEDULING SYSTEM.
                                                                                  31
C
                                                                                  32
      INTEGER F0/8/.F1/9/.
                                  F3/11/,F4/12/,F5/13/
                                                                                  33
      INTEGER#4 SAMSIZ. THELAM, FILEO (4005) . NA(1000)
                                                                                  34
      INTEGER+4 NUDEU(1000), NUDET(1000), NCT(1000), IDIST(1000.5)
                                                                                  35
      INTEGER*2 TIME(1000.6), COST(1000.6), TEST1. TEST2. TEST3. STIME
                                                                                  36
      REAL*4 PARM1(1000.5).PARM2(1000.5).PARM3(1000.5).PARM4(1000.5)
                                                                                  17
      REAL 44 THETA(3). LAMBDA(3)
                                                                                  38
      EQUIVALENCE (FILEO(1), NACT). (FILEO(2). NODES). (FILEO(3). NSRCE).
                                                                                  39
           (FILEO(4).NSINK), (FILEO(5), LNODEN), (FILEO(6).NA(1)),
                                                                                  40
           (FILEO(1006) . NODEO(1)) . (FILEO(2006) . NODET(1)).
                                                                                  41
           (FILEO(3006),NCT(1))
                                                                                  42
      COMMON /BLKA/NACT.NODES.NSRCE.NSINK.LNODEN.NA.NODED.NODET.NCT
                                                                                  43
      INTEGER NORM(5)/*NORM*, 'AL*, 3** '/, BET(5)/*BETA*, 4** */
                                                                                  44
      INTEGER RECT(5)/ 'RECT' . 'ANGU' . 'LAR( ' . 'UNIF' . 'ORM) '/
                                                                                  45
      INTEGER FIXED(5)/'FIXE'.'D(CO'.'NSTA'.'NT)'.' '/
                                                                                  45
                                                                                  47
      WRITE(6.2000)
2000 FORMAT(1H1,132(***)/1X,132(***)/////51X,*S T A T I S T I C A L
                                                                                  48
                                                                                  40
     *P E R T'/////)
      MAITE(6.2001)
                                                                                  50
2001 FORMAT (52x . 'A PROJECT SCHEDULING PROCEDURE !///
                                                                                  51
     *60x. * WRITTEN AT THE *//55x. * INSTITUTE OF STATISTICS */
                                                                                  52
     +57x, TEXAS AGM UNIVERSITY //53x, COLLEGE STATION, TEXAS 77843 ///
                                                                                  53
                                                                                  54
     *61x. AUGUST 1978 /////33x. INQUIRIES AND COMMENTS SHOULD BE ADDRES
     *SED TO: ROBERT L. SIELKEN JR. "//1X.132("*")/1X.132("*")/////)
                                                                                  55
                                                                                  56
         GENERATE PARTS A AND B OF FILEO AND FILEI
C
                                                                                  58
      READ (5.110) NACT , NODES , NSRCE , NSINK , L NODEN
                                                                                  59
```

8

```
READ(5.110)NA(1).NODEO(1).NODET(1).NNCT
                                                                                    61
                                                                                    62
      NCT(1)=NNCT
      ANCTI=NNCT-1
                                                                                    63
      READ(5.110)(TIME(1.J).J=1.NNCT)
                                                                                    64
      READ(5.110)(COST(1.J).J=1.NNCT)
                                                                                    65
      DO 1300 L=1.NNCT1
                                                                                    66
      READ(5.150)J.IDIST(1.J).PARM1(1.J).PARM2(1.J).PARM3(1.J).PARM4(1.J
                                                                                    67
     .)
                                                                                    68
1300 CONTINUE
                                                                                    69
      BRITE(FO)FILEO
                                                                                    70
                                                                                    71
      READ(5.110) TEST1. TEST2. JMAT
      WRITE(FO) JMAT
                                                                                    72
      ENDFILE FO
                                                                                    73
      TEST3=0
                                                                                    74
      STIME=0
                                                                                    75
      BRITE(FI)TESTI.TEST2.TEST3.STIME
                                                                                    76
      DO 1400 I=1.NACT
                                                                                    77
      NACT=NCT(I)
                                                                                    78
      NNCT1=NNCT-1
                                                                                    79
      WRITE(F1)(TIME(1.J).J=1.NNCT).(COST(1.J).J=1.NNCT)
                                                                                    80
      DG 1400 J=1.NNCT1
                                                                                    81
      write(f1)IDIST(1.J).PARM1(1.J).PARM2(1.J).PARM3(1.J).PARM4(1.J)
                                                                                    82
1400
     CENTINUE
                                                                                    83
      ENDFILE FI
                                                                                    84
C
                                                                                    85
C
         GENERATE PART A OF FILE3
                                                                                    86
C
                                                                                    87
      READ(5.110) IEUF . NMAX . IPODL . SAMSIZ
                                                                                    88
      READ(5,112)THELAM. (THE TA(1).1=1.3). (LAMBDA(1).1=1.3)
                                                                                    89
      WRITE(F3) IEDF, NMAX, IPOCL, SAMSIZ, THELAM, (THETA(1), 1=1,3), (LAMBDA(1)
                                                                                    90
     *.I=1.3)
                                                                                    91
C
                                                                                    92
         GENERATE PART A OF FILE4 AND PART A OF FILE5
                                                                                    93
C
                                                                                    94
      READ(5.150) LOPT.NT.PCT.PD.TT
                                                                                    95
      NC YC=1
                                                                                    96
      NFLAG=0
                                                                                    97
      BRITE(F4) NCYC . TT . NFLAG
                                                                                    98
      WRITE (F5) IOPT.NT.PCT.PD
                                                                                    99
110
      FORMAT (8110)
                                                                                   100
112
      FORMAT(110,6F10.5)
                                                                                   101
150
      FCFMAT (2110.4F10.5)
                                                                                   102
C
                                                                                   103
         OUTPUT THE PROBLEM AS IT WAS READ IN
C
                                                                                   104
C
                                                                                   105
      BRITE(6.1000)
                                                                                   106
     FCRMAT( OTHIS IS THE DUTPUT FROM THE MAIN PROGRAM: MAIN !///
                                                                                   107
     * OTHE PROJECT NETWORK AS IT WAS READ IN: 1)
                                                                                   108
      WRITE (6.1001) NUDES . NACT . NSRCE . NSINK . L NODEN
                                                                                   109
1001 FORMAT(/// OTHE NUMBER OF NODES IS ". I4. ". "/ THE NUMBER OF ACTIVI
                                                                                   110
     *TIES IS ".14."." THE SOURCE NODE IS NUMBERED ".15." AND THE SINK
                                                                                   111
     . NCDE IS NUMBERED .. IS. .. . THE LARGEST NODE NUMBER IS ..
                                                                                   112
     *15. . . . // · IACTIVITY ORIG TERM J TIME COST DISTRIBUTION . . 9X. . PARAMET
                                                                                   113
     *ERS*/)
                                                                                   114
      DO 1100 I=1.NACT
                                                                                   115
      NNCT=NCT(1)
                                                                                   116
      NNCT1=NNCT-1
                                                                                   117
      DO 1090 J=1.NNCT1
                                                                                   118
      IC=1015T(1.J)+2
                                                                                   119
      GC TC (1020,1040,1060,1080).1D
                                                                                   120
     IF (PARM4(1.J).EQ.-1.)GO TO 1030
                                                                                   121
```

```
#RITE(6.1021) I . NODEO(1) . NODET(1) . J.TIME(1. J) . COST(1. J) .
                                                                               122
        BET.PARM1(1.J).PARM3(1.J).PARM2(1.J).PARM4(1.J)
                                                                               123
1021 FCRMAT(3x.14.2x.15. 15.1x.11.15.15.1x.5A4.1x.
                                                                               124
     **MIN='.F13.5.'. MAX=',F15.5.'. ALPHA=',F12.5.'. BETA='.F12.5)
                                                                               125
     GC TC 1090
                                                                               126
1030 WRITE(6.1031)1.NODEO(1).NODET(1).J.TIME(1.J).COST(1.J).
                                                                               127
     #BET.PARMI(I.J).PARM2(I.J).PARM3(I.J)
                                                                               128
1031 FCFMAT(3x.14.2x.15. I5.1x.11.15.15.1x.5A4.1x.
                                                                               129
     *'MIN=",F13.5.", MUDE=",F14.5.", MAX=",F14.5)
                                                                               130
     GO TO 1090
                                                                               131
1040 IF (PARM4(I.J).EQ.-1.)GO TO 1050
                                                                               132
     wRITE(6.1041)1.NODEO(1).NODET(1).J.TIME(1.J).COST(1.J).
                                                                               133
        NORM.PARM2(I.J).PARM3(I.J)
                                                                               134
1041 FORMAT(3x,14.2x,15. I5.1x,11.15.15.1X,5A4.1X.
                                                                               135
     4 "MEAN= ", F12.5, ", ST.DEV= ", F12.5)
                                                                               136
     GC TC 1090
                                                                               137
1050 WRITE(6.1051)1.NODEO(1).NODET(1).J.TIME(1.J).CGST(1.J).
                                                                               138
     NCRM,PARM1(I,J),PARM2(I,J),PARM3(I,J)
                                                                               139
1051 FORMAT(3X.14.2X.15. 15.1X.11.15.15.1X.5A4.1X.
                                                                               140
     **MIN=".F13.5.", MEAN=".F14.5.", MAX=".F14.5)
                                                                               141
     GO TO 1090
                                                                               142
1060 WRITE(6,1061)I,NODEO(1),NODET(1),J,TIME(1,J),COST(1,J),
                                                                               143
     * RECT.PARMI(I.J).PARM3(I.J)
                                                                               144
1061 FCRMAT(3X.14,2X.15. I5.1X.[1.[5.[5.1X.5A4.1X.
                                                                               145
    * "MIN= " ,F13.5.", MAX= " ,F15.5)
                                                                               146
     GO TO 1090
                                                                               147
1080 MRITE(6.1081)I.NODEO(I).NODET(I).J.TIME(I.J).COST(I.J).
                                                                               148
        FIXED.PARMI([.J)
                                                                               149
1081 FCRMAT(3x,14.2x.15. I5.1x.11.15.15.1x,5A4.1x, 'TIME=',F12.5)
                                                                               150
1090 CENTINUE
                                                                               151
      JENNCT
                                                                               152
      walte(6,1095)1,NODEO(1),NODET(1),J,TIME(1,J),COST(1,J)
                                                                               153
1095 FORMAT(3x,14,2x,15, 15,1x,11,15,15)
                                                                               154
1100 CONTINUE
                                                                               155
      IF ( TEST 1) 1104.1104.1106
                                                                               166
1104 WRITE(6.1105)
                                                                               157
1105 FORMAT(////*OTHE INFORMATION INPUT TO THE DETERMINISTIC PROJECT SC
                                                                               158
     *HEDULING PRUGRAM WILL BE PRINTED. 1)
                                                                               159
                                                                               160
:106 MRITE(6.1107)
                                                                               161
1107 FERMAT(////*OTHE INFORMATION INPUT TO THE DETERMINISTIC PROJECT SC
                                                                               162
     *HEDULING PROGRAM WILL NOT BE PRINTED. 1)
                                                                               163
1110 IF (TEST2) 1114 . 1114 . 1116
                                                                               164
1114 WRITE(6,1115)
                                                                               165
1115 FORMAT( * THE INTERMEDIATE OUTPUT GENERATED BY THE DETERMINISTIC PR
                                                                               166
     OJECT SCHEDULING PROGRAM WILL BE PRINTED. )
                                                                               167
      GO TO 1120
                                                                               168
1116 WRITE(6,1117)
                                                                               169
1117 FCRMAT( THE INTERMEDIATE CUTPUT GENERATED BY THE DETERMINISTIC PR
                                                                               170
     *OJECT SCHEDULING PROGRAM WILL NOT BE PRINTED. 1)
                                                                               171
                                                                               172
1120 IF (JMAT) 1128.1128.1121
                                                                               173
1121 WRITE(6,1122)
                                                                               174
1122 FCRMAT("OTHE SIMPLIFICATION PROGRAM WILL PRINT CNLY THE MINIMUM CU
                                                                               175
                                                                               176
     OTPUT . 1)
                                                                               177
      GC TO 1130
1128 WRITE(6,1129)
                                                                               178
1129 FORMAT('OTHE SIMPLIFICATION PROGRAM WILL PRINT THE COMPLETE OUTPUT
                                                                               179
                                                                               180
     ....
1130 CENTINUE
                                                                               181
                                                                               182
      WRITE(6.1131) LEDF . NMAX . SAMSIZ
```

```
FORMAT ( ODURING THE SUBNETWORK ANALYSIS PROGRAM, 1/
                                                                              183
     *5x. THE NUMBER OF SUBDIVISIONS IN THE ESTIMATED DURATION DISTRIBUT
                                                                              184
     *ION FOR EACH SUBNETWORK WILL BE ".14.":"/
                                                                              185
     *5X. "THE LARGEST NUMBER OF ACTIVITIES IN A SUBNETWORK FOR WHICH TH
                                                                              186
     187
     *ILL BE ENUMERATED IS ". 14. "; "/
                                                                              188
          *THE MAXIMUM NUMBER OF ACTIVITY DURATION CONFIGURATIONS EXPLI
     *5 X .
                                                                              IRG
     *CITLY CONSIDERED IN THE DETERMINATION OF THE */10x, UPPER AND LOWER
                                                                              190
     * BOUNDS ON EACH SUBNETWORK . 14 . 'S DISCRETE DURATION DISTRIBUTION
                                                                               191
                                                                              192
     *WILL BE ',15,";")
      IF(IPCOL)1132,1132,1134
                                                                              193
1132 WRITE(6.1133)
                                                                              194
1133 FORMAT (5x, THE MAXIMUM-CLUSTER PROCEDURE WILL BE USED IN THE DETER
                                                                              195
     *MINATION OF THE BOUNDS: 1)
                                                                              196
                                                                              197
      GO TO 1136
1134 WRITE(6.1135)
                                                                              198
1135 FORMAT(5X. THE UNION-CLUSTER PROCEDURE WILL BE USED IN THE DETERMI
                                                                              199
     ANATION OF THE BOUNDS; 1)
                                                                              200
1136 WRITE(6.1137)
                                                                              201
1137 FORMAT (5x, THE FOLLOWING (THETA, LAMBCA) PAIRS WILL BE USED IN THE
                                                                              202
     *FORMATION OF THE CLUSTERS: 1)
                                                                              203
      1F(THELAM)1138,1138,1140
                                                                              204
1138
     WRITE(6.1139)
                                                                              205
                      1.00000. 1.00000).( 2.00000. 2.00000).(
1139 FORMAT(10X. 1
                                                                              206
     *00000, 2.000001.*1
                                                                              207
      GO TO 1150
                                                                              208
1140 WRITE(6.1141)(THETA(1).LAMBDA(1).[=1.3)
                                                                              209
1141 FGRMAT(10X.*(*,F10.5.*,*,F10.5.*).(*,F10.5.*.*,F10.5.*).(*,F10.5.
                                                                              210
     **.*.F10.5.*).*)
                                                                              211
1150 WRITE (6.1151) NT
                                                                              212
1151 FORMAT( OTHE NUMBER OF SUBDIVISIONS IN THE NETWORK .. 14. S SYNTHES
                                                                              213
     *IZED COMPLETION TIME DISTRIBUTION WILL BE "12. ...)
                                                                              214
      IF(IOPT)1160,1160,1165
                                                                              215
1160 WRITE(6.1161)PCT.PD
                                                                              216
1161 FORMAT( OTHE .FT. 2, -TH PERCENTILE OF THE NETWORK .1H. S APPROXI
                                                                              217
     *MATE COMPLETION TIME DISTRIBUTION WILL BE COMPARED TO THE !/
                                                                              218
     *10X, *SPECIFIED PROJECT DEADLINE OF *, F10.5. .. )
                                                                              219
      GO TO 1170
                                                                              220
1165 MRITE(6.1166)PD
                                                                              221
1166 FORMAT (OTHE MEAN OF THE NETWORK OTH OF APPROXIMATE COMPLETION TI
                                                                              222
     *ME DISTRIBUTION WILL HE COMPARED TO THE 1/
                                                                              223
     *10X. *SPECIFIED PROJECT DEADLINE OF *.F10.5. .. )
                                                                              224
1170 WRITE(6.1171)TT
                                                                              225
1171 FORMAT( OTHE TARGET TIME FOR THE FIRST ITERATION IS .. F10.5. .. //)
                                                                              226
                                                                              227
         CHECK FOR LOOPS IN THE PROJECT NETWORK
C
                                                                              228
c
                                                                              229
      CALL LOOP
                                                                              230
      STOP
                                                                              231
      ENG
                                                                              232
      SUBREUT INE LOOP
                                                                              233
                                                                              234
         THIS PROGRAM DETERMINES WHETHER THE PROJECT NETWORK IS ACYCLIC
£
                                                                              235
         DE CONTAINS NODES WHICH ARE PART OF LOOPS (CYCLES)
×
                                                                              236
                                                                              237
      THRE ICIT INTEGERS (A-Z)
                                                                              238
      COMPASSOR ALTERVAL ALTERVAL
                                                                              239
      BART 3-7 100 MATE 0981, TATL 1 1000 | . HEAD(1 000) . NCT(1000)
                                                                              240
     CONSIST FOR SEASON, SERVER, NEST ME .L. HODEN . NA . TAIL . HE AD . NCT
                                                                              241
                                                                              242
                                                                              243
```

```
WRITE(6.2004)
                                                                                  244
2004 FGFMAT (*OTHIS IS THE OUTPUT FROM SUBPROGRAM LOOP. THE PROGRAM SEAR
                                                                                  245
     *CHES THE GIVEN PROJECT NETWORK FOR LOOPS (CYCLES).*/* A VALID ..
                                                                                  246
     **PROJECT NETWORK SHOULD CONTAIN NO LOGPS.*///)
                                                                                  247
      DO 4 J=1.LNODEN
                                                                                  248
      0=(L)A
                                                                                  249
      B(J)=0
                                                                                 250
C
      FORM THE 1ST HIERARCHY
                                                                                 251
      INCOE = 0
                                                                                 252
   80 HIER = 2
                                                                                 253
82
      INCOE=INCCE+1
                                                                                 254
      If (INCDE.EQ.NSRCE.OR.INODE.EQ.NSINK) GO TO 81
                                                                                 255
      DO 5 1=1.M
                                                                                 256
      IF (INODE.EQ.TAIL(I))GO TO 81
                                                                                 257
      CONTINUE
5
                                                                                 258
      GO TO 82
                                                                                  259
      J=0
21
                                                                                 260
      DO 1 1= 1.M
                                                                                 261
      IF (HEAD(I).NE. INCOE) GO TO 1
                                                                                 262
      J = J+1
                                                                                 263
      A(J)=TAIL(I)
                                                                                 264
      IF (TAIL(1).EQ. INODE) GO TO 998
                                                                                 265
    1 CONTINUE
                                                                                 266
      IF (J.EQ.0) GO TO 997
                                                                                 267
      LASJ
                                                                                 268
      J=O
                                                                                 269
      FORM THE SUBSEQUENT HIERARCHIES
                                                                                 270
  102 CONTINUE
                                                                                 271
      DO 2 11=1.1A
                                                                                 272
      DO 3 1=1.M
                                                                                 273
      IF (HEAD(I).NE.A(II)) GO TO 3
                                                                                 274
      IF(TAIL(1).EQ.INODE) GO TO 998
                                                                                 275
      IF(J.EQ.0) GO TO 40
                                                                                 276
      DO 10 K=1.J
                                                                                 277
      IF(TAIL(1).EU.B(K)) GO TO 11
                                                                                 278
10
      CONTINUE
                                                                                 279
      CONTINUE
                                                                                 280
40
      J=J+1
                                                                                 261
      B(J)=TAIL(I)
                                                                                 282
      CONTINUE
                                                                                 283
    3 CONTINUE
                                                                                 284
    2 CONTINUE
                                                                                 285
      IF (J.EQ.0) GO TO 997
                                                                                 286
      HIER = HIER+1
                                                                                 287
      IA=J
                                                                                  288
      J = 0
                                                                                 289
      DO 20 1=1.1A
                                                                                 290
20
      A(1)=E(1)
                                                                                 291
      GO TO 102
                                                                                 292
  997 CONTINUE
                                                                                 293
                                                                                 294
      WRITE(6.2002) INODE
 2002 FORMAT ( NODE . 15. IS NOT PART OF A LOOP )
                                                                                 295
      IF (INODE.NE.LNODEN) GO TO 80
                                                                                 296
      IF(ILCOP.EQ.1) GO TO 50
                                                                                 297
      BRITE(6.1000)
                                                                                 258
 1000 FCRMAT ( THERE ARE NO LOOPS IN THIS NETWORK !)
                                                                                 299
      GC TC 999
                                                                                 300
      WRITE(6.51)
                                                                                 301
  50
      FORMAT( THERE ARE NO OTHER LOOPS IN THIS NETWORK !)
51
                                                                                 302
      A(1000000000) = 0.
                                                                                 303
      GO TO 999
                                                                                 304
```

1001	B WRITE (6.1001) INDDE.HIER FORMAT(* NODE *.15.*IS PART OF A	1000 THE	E ADE	1.13.1	ACTIVITIE
	+S IN THE LOOP.)	LOUPE THEK			ACTIVITIE
	ILCCP=1				
	INDOD=INGDE+1				
	IF (INCOO.NE.LNODEN) GO TO 80				
999	RETURN				
	ENC				

2. DPS

```
C
      DETERMINISTIC PROJECT SCHEDULING PROGRAM
    DETERMINISTIC PROJECT SCHEDULER: THIS PROGRAM COMPUTES THE
                                                                               3
     MINIMUM PROJECT COST AND ASSOCIATED OPTIMAL FIXED ACTIVITY TIMES
C
     FOR EVERY FEASIBLE PROJECT DEADLINE.
                                                                               5
                                                                               6
        ALL VARIABLES ARE INTEGER.
C
      (IF ANY VARIABLE IS NOT ALREADY IN INTEGER FORM. THE VALUES MUST
                                                                               8
C
      BE RESCALED - THAT IS. MULTIPLIED BY AN APPROPRIATE POWER OF 10 -
      UNTIL THE VALUES ARE INTEGER.)
                                                                              10
                                                                              11
12
                                                                              13
      DEFINITION OF VARIABLES:
C
                                                                              15
         ABAR(I,J) = TIME(I,NK(I)+I-J) + XNODE(ORIG(I))-XNODE(TERM(I))
                                                                              16
            C(1.J) = DECREASE IN I TH ACT'S COST PER UNIT FOR J TH TIME
                                                                              17
              CAP = MIN(FLOW REACHING ORIGIN NODE. EXCESS CAPACITY TO
                                                                              18
                         TERMINAL NODE)
                                                                              19
        CGST(I.J) = COST OF COMPLETING ACTIVITY I AT TIME(I.J)
                                                                              20
               DEL = MIN(DELTA1.DELTA2)
                                                                              21
            DELTA1 = MIN(-ABAR(I.J) WITH I LABELED AND J UNLABELED.
                                                                              22
                        ABAR(1.J)<0)
                                                                              23
            DELTA2 = MIN(ABAR(I.J) WITH I UNLABELED AND J LABELED.
                        ABAR(1.J)>0)
                                                                              25
         DIREC(J) = DIRECTION OF FLOW REACHING NODE J
                                                                              26
                        (0=FCRWARD. 1=REVERSE)
                                                                              27
        FLOW(I.J) = FLOW IN J TH PIECE OF ACTIVITY I
                                                                             28
               INF = ANY NUMBER GREATER THAN MAX(CAP)
                         (CURRENTLY SET AT (2+MAX +1))
                                                                              30
            K1(1) = THE NUMBER OF THE TIME-COST PIECE USED IN
                        LABELING TERM(I) FROM ORIG(I)
                                                                              32
            KOUNT = KEEPS TRACK OF ORDER IN WHICH NODES WERE LABELED
                                                                             33
C
          LABEL(1) = 0 IF NODE I UNLABELED
                                                                              34
                    1 IF NODE I LABELED
                                                                              35
            LINPUT = SPECIFIC VALUE OF LAMBOA IF TEST3=1 OR 2
C
                                                                              36
               NA = TOTAL NUMBER OF ACTIVITIES
                                                                              37
             NK(I) = NUMBER OF DIFFERENT TIMES AND COSTS FOR ACTIVITY I
                                                                              38
               NN = TOTAL NUMBER CF NCDES
                                                                              39
           GRIG(1) = ORIGIN NODE FOR ACTIVITY 1
                                                                              40
          CRIG2(1) = WHERE THE FLOW IS FROM - USED IN LABELING ONLY
                                                                              41
            PCOST = PROJECT COST FUNCTION
                                                                              42
              SINK = NUMBER OF THE SINK NGDE
                                                                              43
            SOURCE = NUMBER OF THE SOURCE NODE
                                                                              44
           TERM(I) = TERMINAL NODE FOR ACTIVITY I
                                                                              45
            TEST1 = OPTION TO SUPPRESE PRINTING OF INPUT
C
                                                                              46
                         ( 0=PRINT. 1=NO PRINT)
                                                                              47
            TEST2 = OPTION TO SUPPRESS INTERMEDIATE OUTPUT
C
                                                                              48
                         ( 0=PRINT. 1=NO PRINT)
                                                                              49
             TEST3 = OPTION TO SPECIFY VALUE FOR LAMBDA
                         (0=NO. 1=YES AND SEE INTERMEDIATE OUTPUT.
                                                                              51
                         2=YES BUT NO INTERMEDIATE GUTPUT)
         TIME(1.J) = J TH BREAKPOINT (DURATION TIME) FOR ACTIVITY I
                                                                             53
          XACT(1) = ACTIVITY DURATION TIME
                                                                             54
          XNUDE(I) = NODE TIME
                                                                              55
          XDIFF(1) = XNODE(ORIG(1))-XNODE(TERM(1)). AN UPPER BOUND ON
                                                                             56
                         THE ACTIVITY DURATION TIME
       I.J.K.M.N.P = INDICES
                                                                             58
      INCDE. ITERM. IACT. IDRIG. ICIFF. ETC.
                                                                             59
                   = NON-INDEXED VERSIONS OF XNODE(I).TERM(I).XACT(I).
                                                                             60
```

```
ORIG(1).XDIFF(1).ETC.
                                                                            61
c
                                                                            62
                                                                            63
64
C
C
        FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS. LET
                                                                            65
        MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL
                                                                            66
C
C
                    PROJECT NETWORK
                                                                            67
        LAMAX = THE LARGEST NODE NUMBER ALLOWED IN THE PROJECT NETWORK
C
                                                                            68
        BKMAX = THE MAXIMUM NUMBER OF BREAK POINTS ALLOWED IN THE
                                                                            69
C
                    ENTIRE PROJECT'S TIME-COST CURVE
                                                                            70
C
                                                                            71
        NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR
C
C
                    EACH ACTIVITY
                                                                            72
C
        CURRENTLY, MMAX=1000; LNMAX=9999; BKMAX=3000; NCTMAX=6
                                                                            73
C
C
     DIMENSION CAP(LNMAX).FLOW(MMAX,NCTMAX).C(MMAX.NCTMAX).GRIG(MMAX).
                                                                            75
C
               TERM(MMAX).TIME(MMAX.NCTMAX).CGST(MMAX.NCTMAX).NK(MMAX).
                                                                            76
C
               ABAR(MMAX, NCTMAX). XDIFF(MMAX). XNODE(LNMAX). XACT(MMAX).
                                                                            77
C
               DIREC(LNMAX).LABEL(LNMAX).K1(LNMAX).ORIG2(LNMAX).
                                                                            78
C
               KOUNT(LNMAX).AORD(MMAX).ND(LNMAX).NDD(LNMAX).IP(MMAX).
                                                                            79
C
               CTIME(MMAX).FILEO(BKMAX).FILEG(2.MMAX).NUMS(MMAX)
                                                                            80
C
                                                                            81
82
                                                                            83
C
      IMPLICIT INTEGER + 2(A-Z)
      INTEGER#4 NA.NN.ORIG.TERM.NK.I.SOURCE.SINK.NUMS(1000).LNDDEN
                                                                            A5
     REAL*4 CAP(9999) .FLOW(1000, 6) .C(1000, 6) .PCGST .INF .PCGST1.TT.
                                                                            86
     IKCCST . ACCST . PNEW
                                                                            87
     CCMMCN ORIG.TERM.NK.NA.LNCDEN.TIME.CTIME, XNODE.ACRD.LMIN.LMAX.
                                                                            88
            TEST1
                                                                            89
     DIMENSION DRIG(1000).TERM(1000).TIME(1000. 6).COST(1000. 6).
                                                                            90
    *NK(1000).ABAR(1000. 6).XDIFF(1000).XNDDE(9999).XACT(1000).
                                                                            91
     *K1(9999),ORIG2(9999),KUUNT(9999),AORD(1000),CTIME(1000).
                                                                            92
    *DIREC(9999).LABEL(9999).ND(9999).ND0(9999).IP(1000)
                                                                            93
      INTEGER F0/8/.F1/9/.
                                                                            94
      DIMENSION FILEO(2000), FILE6(2,1000)
                                                                            95
      WRITE(6.1500)
                                                                            96
1500 FORMAT(1H1,132(***)/1X,132(***)/*OTHIS IS THE OUTPUT FROM THE DETE
                                                                            97
    *RMINISTIC PROJECT SCHEDULER: DPS*//1x.132(***)/1x.132(***))
                                                                            98
C
                                                                            99
     INPUT DATA
C
                                                                           100
                                                                           101
     REMIND FO
                                                                           102
     REWIND F1
                                                                           103
                                                                           104
     READ (FO) NA.NN. SOURCE.SINK.LNODEN.NUMS.ORIG.TERM.NK
                                                                           105
                                                                           106
     PCCST=0.
                                                                           107
     READ(F1) TEST1. TEST2. TEST3.LI NPUT
                                                                           108
      DO 12 I=1.NA
                                                                           109
     KN=NK(I)
                                                                           110
     READ(F1)(TIME(1.J).J=1.KN).(COST(1.J).J=1.KN)
                                                                           111
     KN1=KN-1
                                                                           112
     DO 12 J=1.KN1
                                                                           113
     REAC(F1)
                                                                           114
     CONTINUE
                                                                           115
      IF(TEST1.E0.1) GO TO 401
                                                                           116
      WRITE(6.150) NN.NA. SOURCE.SINK
                                                                           117
401
     CALL CRDER
                                                                           118
                                                                           119
C
      SET UP INITIAL VALUES
                                                                           120
                                                                           121
```

	1F(TEST1.EQ.1) GC TO 193	122
	K3=1	123
192	K2=K3+8	124
	IF(K2.GT.LNGDEN)K2=LNGDEN	125
	bRITE(6.151) (K.K=K3.K2)	126
	WRITE(6.157) (XNODE(K).K=K3.K2)	127
	IF(K2.GE.LNUDEN)GO TO 191	128
	K3=K2+1	129
	GO TO 192	130
191	MRITE(6,152)	131
193	DC 10 I=1.NA	132
	LABEL(I)=0	133
	XDIFF(I)=XNCDE(ORIG(I))-XNCDE(TERM(I))	134
	NKM1=NK(I)-1	135
	Kn=nk(1)	136
	DO 9 J=1,NKMI	137
	TT=TIME(I,J+1)-TIME(I,J)	138
	IF(TT)7,8,7	139
7	C(1.J)=(C05T(1.J)-C05T(1,J+1))/TT	140
	GC TC 6	141
	C(1.J)=0.	142
6	IF(INF.LT.C(I,J)) INF=C(I,J)	143
	XACT(I)=XDIFF(I)	144
	IF(XACT(1)-LT-T[ME(1-J+1)) XACT(1)=T[ME(1-J+1)	145
	JJ=NK(I)-J+1	146
	ABAR(I, J)=TIME(I, JJ)+XDIFF(I)	147
	FLOW(I.J)=0 CONTINUE	149
•	FILE6(1.1)=0	150
	F1LE6(2,1)=XACT(1)	151
	ABAR(I,KN)=TIME(I,1)+XDIFF(I)	152
	FLC*(I,KN)=0	153
	IF(TEST1.EQ.1) GO TO 10	154
	WRITE(6,153)NUMS(1), XACT(1), GRIG(1), TERM(1), (J,TIME(1,J),COST(1,J	-
	*.C([.J).ABAR([.J).J=1.NKM])	156
	BRITE(6.156) KN.TIME(I.KN).COST(I.KN).ABAR(I.KN)	157
10	CENTINUE	158
	WRITE(F6)((FILE6(I.J).1=1.2).J=1.NA)	159
	INF=2.*INF+1.	160
	DO 417 I=1.NA	161
	C(I, hk(I))=0.	162
	NKM1=NK(I)-1	163
	PCCST1=0.	164
	IKK=0	165
	DO 418 K=1.NKM1	166
	IF(K.NE.1) GO TO 40	167
	XIJ=XACT(I)	168
	IF(XIJ.GT.TIME(1.2)) XIJ=TIME(1.2)	169
	GO TO 41	170
40	XIJ=XACT(1)-TIME(1,K)	171
	<pre>IF(XIJ.LT.0) XIJ=0 IF(XIJ.GT.(TIME(I.K+1)-TIME(I.K))) XIJ=TIME(I.K+1)-TIME(I.K)</pre>	172
	IF(IKK.EG.1) GO TO 41	173 174
	IF(C(1,K).GT.C(1,K-1)) GO TO 50	175
41	PCGST1=PCGST1+C(I.K)+XIJ	176
٠,	GC TO 418	177
50	IKK=1	178
30	MRITE(6,237) 1.1	179
	PCOSTI=PCOSTI+C(I.K)+XIJ	180
418	CUNTINUE	181
	PCCST=PCOST+COST(I.1)+C(I.1)+TIME(I.1)-PCOST1	182

		PNEN=PCCST	183
	417	CCNTINUE	184
		LAMBDA=LMAX	185
		KKKK=KKKK+1	186
		FILEO(KKKK)=LAMBDA	187
		IF (TEST3.GE.1) GO TO 700 WRITE(6.154)	189
		LINPUT=0	190
		GC TC 96	191
70	0	CCNTINUE	192
		IF(LINPUT.LT.LMIN) GO TO 705	193
		IF(LINPUT.GE.LMAX) GG TO 704	194
		IF (TEST3.EQ.2) GO TO 724	195
	~	WRITE(6.155) LINPUT	196
	90	WRITE(6.200) LAMBDA, PCOST IF(TEST2.EQ.1.DR.TEST3.GE.1) GO TO 724	198
		WRITE(6,235)	199
	724	CAP(SOURCE)=INF	200
		ITER=0	201
	99	LABEL(SGURCE)=1	202
		IF(TEST2.EQ.1.UR.TEST3.GE.1) GO TO 97	203
		ITER=ITER+1	204
		WRITE(6,225) ITER	205
C		INTERNAL AND INC. ITERATION	206 207
C		INITIAL LABELING ITERATION	207
•	97	I=1	209
		J=SOURCE	210
		M=0	211
C		IF ACTIVITY STARTS AT DESIGNATED ORIGIN. TRY TO LABEL.	212
C		OTHERWISE. CHANGE ORIGINS.	213
	14	IF (ORIG(I).NE.J) GO TO 13	214
c		ITERM=TERM(I) CHECK IF NODE ALREADY LABELED AND	215
c		CHECK IF ABAR(1.NK(1))=0.	217
•		IF (LABEL(ITERM).NE.O.OR.ABAR(I.NK(I)).NE.O) GO TO 13	218
c		IF NODE NOT ALREADY LABELED AND ABAR(I.NK(I))=0.	219
C		PROCEDE WITH LABELING.	220
		LABEL (ITERM)=1	221
		ORIG2(ITERM)=J	222
		K1(ITERM)=NK(I)	223
		DIREC(ITERM)=I CAP(ITERM)=INF	224
		IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 403	226
		WRITE(6,201) ITERM.ORIG2(ITERM).KI((ITERM)	227
C		IF CAN REACH SINK. TERMINATE (IMPLIES INFEASIBLE)	228
	403	IF (ITERM.EQ.SINK) GO TO 15	229
		M=M+1	230
		KCUNT(M)=ITERM	231
C		IF EVERY PATH TESTED AND INFINITE FLCW NOT POSSIBLE.	232
	12	GO ON TO LABELING PART(II). 1=1+1	233
	• •	I=I+1 IF (I+GT+NA) GD TO 11	235
		GC TC 14	236
c		CHANGE DESIGNATED ORIGINS.	237
	11	IF (J.EQ.SOURCE) P=1	238
		1F(P.GT.M) GO TO 16	239
c		IF ALL LABELED NODES HAVE BEEN SCANNED AND NO NEW NODES	240
C		HAVE EEEN LABELED. GO ON TO LABELING PART (II). J=KOUNT(P)	241
		D=P+1	243
			-43

```
I=1
GO TO 14
                                                                                 244
                                                                                 245
   15 IF(TEST3.GE.1) GO TO 404
                                                                                 246
      WRITE(6,202) LAMBDA
                                                                                 247
  404 GO TO 999
                                                                                 248
   16 IF (TEST2.EQ.1.OR.TEST3.GE.1) GO TO 405
                                                                                 249
      WRITE(6,203)
                                                                                 250
C
                                                                                 251
C
     NEXT LABELING PROCEDURE
                                                                                 252
                                                                                 253
  405 I=1
                                                                                 254
      J=SQUECE
                                                                                 255
C
      AGAIN. CHECK ALL CONDITIONS FOR LABELING
                                                                                 256
C
      IE. CHECK IF NODE IS ALREADY LABELED. IF ABAR(I.J)=0. AND
                                                                                257
C
      IF THE FLOW(I.J) IS LESS THAN ITS UPPER BOUND.
                                                                                 258
   20 IF (ORIG(1).NE.J) GO TO 24
                                                                                 259
      ITERM=TERM(I)
                                                                                 260
      KN=NK(I)
                                                                                 261
      DO 25 K=1.KN
                                                                                 262
      IF (K.EQ.KN) GO TO 27
                                                                                 263
      IF(LABEL(ITERM).NE.O.OR.ABAR(I.K).NE.O.OR.FLOW(I.K).GE.
                                                                                264
     1(C(I.NK(I)-K)-C(I.NK(I)-K+1)))GO TO 25
                                                                                 265
      DIREC(ITERM)=I
                                                                                266
      CAPACITY IS MIN OF PREVIOUS FLOW AND THE EXCESS CAPACITY
                                                                                 267
      CAP(ITERM)=C(I,NK(I)-K)-C(I,NK(I)-K+1) - FLOW(I,K)
                                                                                 268
      GO TO 23
                                                                                269
   27 IF(LABEL(ITERM).NE.O.OR.ABAR(I.K).NE.O.OR.FLOW(I.K).GE.INF)
                                                                                 270
     1 GO TO 25
                                                                                 271
      IF THE NOCE HAS NOT ALREADY BEEN LABELED, ABAR(I,J)=0, AND
                                                                                 272
C
      THE FLOW IS LESS THAN ITS UPPER BOUND, PROCEDE WITH THE LABELING
                                                                                 273
      OF THE NODE.
                                                                                 274
      DIREC(ITERM)=1
                                                                                 275
      CAP(ITERM)=INF
                                                                                 276
   23 LABEL(ITERM)=1
                                                                                 277
      ORIG2(ITERM)=J
                                                                                 278
      K1(ITERM)=K
                                                                                279
      IF (CAP(ITERM).GT.CAP(ORIG(I))) CAP(ITERM)=CAP(ORIG(I))
                                                                                 280
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 406
                                                                                261
      wRITE(6.204) ITERM, ORIG2(ITERM), KI(ITERM), DIREC(ITERM), CAP(ITERM)
                                                                                282
C
      IF SINK LABELED. GO TO UPDATE PROCEDURE
                                                                                283
  406 IF (ITERM.EQ.SINK) GO TO 21
                                                                                284
     TH=M+1
                                                                                 285
      KGUNT (M)= LTERM
                                                                                 286
      CHECK IF ALL PATHS TRIED
                                                                                 287
   25 CONTINUE
                                                                                288
      GO TO 19
                                                                                289
   24 IF(TERM(I).NE.J) GO TO 19
                                                                                290
      IORIG=ORIG(I)
                                                                                291
      KN=NK(I)
                                                                                292
      DO 26 K=1.KN
                                                                                293
      IF(LABEL(IORIG).NE.O.OR.ABAR(I.K).NE.O.OR.FLOW(I.K).LE.O)
                                                                                294
     2 GO TO 26
                                                                                295
      DIREC(IORIG) =- I
                                                                                 296
      CAP(IORIG)=FLOW(I.K)
                                                                                297
      LABEL (IORIG)=1
                                                                                 298
      OFIG2(IORIG)=J
                                                                                299
      KI(LORIG)=K
                                                                                300
      IF(CAP(IORIG).GT.CAP(TERM(I))) CAP(IORIG)=CAP(TERM(I))
                                                                                 301
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 402
                                                                                302
      WRITE(6.204) IURIG.ORIG2(IURIG).K1(IURIG).DIREC(IURIG).CAP(IURIG)
                                                                                303
  402 M=M+1
                                                                                304
```

		KOUNT(M)=10RIG	305
		CONTINUE	306 307
	19	I=I+1	307
		IF (I.GT.NA) GU TO 18 GO TO 20	309
	10	IF (J.EQ.SQURCE) P=1	310
	10	IF (F.GT.M) GO TO 22	311
		J=KOUNT(P)	312
		P=P+1	313
		I=1	314
		GC TO 20	315
C			316
C		NONBREAKTHRUUGH HAS OCCURED. DELTAS ARE FOUND AND UPDATING	317
C		MADE IN THE XNODES AND XACTS.	318
C			319
	22	CELTAI=INF+1	320
		DELTA2=INF+1	321
		DC 4 I=1.NA	322
		KN=NK(I)	323
		<pre>IF (LABEL(ORIG(I)).EQ.1.AND.LABEL(TERM(I)).EQ.0) GO TO 1</pre>	324
C		AT IS SET OF I LABELED AND J UNLABELED.	325
C		A2 IS SET OF I UNLABELED AND J LABELED.	326
		IF (LABEL (CRIG(I)).EQ.O.AND.LABEL (TERM(I)).EQ.1) GO TO 2	327
_		GC TO 4	328
C		FINCING CELTAI'S.	329
		DO 3 J=1.KN	330 331
		IF (ABAR(I.J).GE.O) GO TO 3 IF (-ABAR(I.J).LT.DELTA1) DELTA1=-ABAR(I.J)	332
	2		333
	•	GO TO 4	334
c		FINDING DELTA2'S	335
•	2	DO 5 J=1,KN	336
	-	IF(ABAR(I.J).LE.O) GO TO 4	337
		IF (ABAR(I,J).LT.DELTA2) DELTA2= ABAR(I,J)	338
	5	CONTINUE	339
	4	CONTINUE	340
C		DEL=MIN(DELTA1.DELTA2)	341
		DEL=DELTA1	342
		IF (DELTA2.LT.DEL) DEL=DELTA2	343
		LAMBCA=LAMBDA-DEL	344
		KKKK=KKKK+1	345
		FILEO(KKKK)=LAMBDA	346
C		UPDATING THE XNODES.	347
		IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 407	348
		WRITE(6.206) LAMBDA	349
	407	IF (TEST3.EQ.2) GO TO 721	350
		DELTA= LAMBDA + DEL	351
		MRITE(6,209) DEL.DELTA.LAMBDA.DELTA	352
	721	IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 408	353
		WRITE(6,207)	354
	408	DO 80 I=1,LNODEN	355
		INCDE=XNCDE(I)	356
		IF(LABEL(1).EQ.0) GO TO 81 IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 409	357 358
		WRITE(6,210) 1. INODE	359
	400	XNCDE(I)=INODE	360
	409	GG TC 80	361
	AI	1F(TEST2.EQ.1.QR.TESTJ.GE.1) GO TO 410	362
	٠.	WRITE(6.211) I.INODE	363
	410	XNCDE(I)=INCDE-DEL	364
		CONTINUE	365

36

36

36

37

37

37

37

374

37

37

37

378

379

38

38

38

38

384

385

386

38

388

389

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

415

420

421

422

423

424

425

426

```
UPDATING THE XACTS (3 PARTS) IF (TEST3.EQ.2) GO TO 722
      BRITE(6.212) DELTA
  722 FCGST=0.
      DC 82 I=1.NA
      IP(1)=0
      PCOST1=0.
      NKM1=NK(I)-1
      IACT=TIME(I.NK(I))
      IORIG=CRIG(1)
      ITERM=TERM(I)
      IDIFF=XNGDE(ITERM)-XNODE(IORIG)
      XDIFF(1)=-IDIFF
      IF (IDIFF.GE.IACT) GO TO 86
      XACT(1)=IDIFF
      DO 550 K=1.NKM1
      IF (K.NE.1) GO TO 43
      XIJ=XACT(I)
      IF(XIJ.GT.TIME(1.2)) XIJ=TIME(1.2)
      FLAG1=0
      GO TO 42
   43 XIJ=XACT(I)-TIME(I,K)
      IF(XIJ.LT.0) GO TO 552
      IF(XIJ GT \cdot (TIME(I \cdot K + I) - TIME(I \cdot K))) XIJ = TIME(I \cdot K + I) - TIME(I \cdot K)
      FLAG1=0
      GO TO 42
  552 FLAG1=1
      FLAG2=K-1
      GC TC 553
   42 PCGST1=PCGST1+C(1.K) +XIJ
  550 CENTINUE
  553 KCGST=COST(1.1)+C(1,1)*TIME(1.1)
      ACOST=KCOST-PCOST1
      PCCST=PCOST+ACOST
      IF (LABEL(IORIG)-LABEL(ITERM)) 83.84.85
   83 IDIFF=IDIFF-DEL
      FILE6(1.1)=1
      FILE6(2,1)=101FF
      IF (TEST3.EQ.2) GO TO 82
      IF(FLAG1.EQ.1) GO TO 59
      ACGST=ACOST + C(I.NKM1) +DEL
      IP(1)=1
      WRITE(6,214)NUMS(1), IDIFF, ACOST, C(1, NKM1)
      GO TO 82
   59 ACOST=ACOST+C(I.FLAG2)+DEL
      IP(1)=1
      WRITE(6.214) NUMS(I) . IDIFF . ACCST . C(I.FLAG2)
      GC TC 82
84
      FILE6(1.1)=0
      FILE6(2.1)=XACT(1)
      IF(TEST3.E0.2)GO TO 82
      WRITE(6,216)NUMS(1), XACT(1), ACOST
      GO TC 82
   85 IDIFF=IDIFF+DEL
      FILE6(1.1)=-1
      FILE6(2.1)=IDIFF
      IF(TEST3.EQ.2)GO TO 82
      IF(FLAGI.EQ.I) GO TO 58
      ACOST=ACOST - C(1. NKM1) +DEL
      IP(1)=2
      WRITE(6,213)NUMS(1), IDIFF, ACGST, C(1, NKM1)
```

		GC TC 62	427
	58	ACCST=ACOST-C(1.FLAG2)*DEL	428
		IP(1)=2	429
		BRITE(6.213) NUMS(1). IDIFF. ACOST. C(1. FLAG2)	430
	0.	GO TO 82	431
	00	XACT(I)=IACT DG 551 K=1.NKM1	433
		IF(K.NE.1) GO TO 45	434
		XIJ=XACT(I)	435
		IF(XIJ.GT.TIME(1.2)) XIJ=TIME(1.2)	436
		GC TC 46	437
	45	XIJ=XACT(I)-TIME(I.K)	438
		IF(XIJ.LT.O) XIJ=0	439
		IF(XIJ.GT.(TIME(I.K+1)-TIME(I.K))) XIJ=TIME(I.K+1)-TIME(I.K)	440
	46	PCOST1=PCOST1+C(1.K)*XIJ	441
5	51	CENTINUE	442
		KCGST=CGST([,1)+C([,1)*TIME([,1)	443
		ACGST=KCOST-PCOST1	444
		PCOST=PCCST+ACOST	445
		FILE6(1.1)=0	446
		FILE6(2.1)=XACT(1)	447
		IF (TEST3-EQ-2) GO TO 82	448
		WRITE(6.216)NUMS(I).XACT(I).ACOST	449
	29	CCNTINUE	450
		WRITE(F6)((FILE6(I.J), I=1.2), J=1.NA)	451
		IF (TEST3.EQ.2) GO TO 723	452
		PCOST1=PNEW	453
		PNEW=(PCGST-PNEW)/DEL	454
		WRITE(6.224) PCOSTI.PNEW	455
7	23	PNEW=PCCST	456
		IF (TEST3.NE.O.AND.LAMBDA.LE.LINPUT) GO TO 703	457
C		RESET LABELS TO 0 AND REFIGURE ABARS.	458
C		THEN START OVER.	459
		DC 87 I=1,LNGDEN	460
		LABEL(1)=0	461
	87	CONTINUE	462
		IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 420 WRITE(6.226) (J.J=1.11)	463
	20	DO 88 I=1.NA	465
•	20	NKM1=NK(I)-1	466
		DO 500 K=1.6KM1	467
		J=NKM1+2-K	468
5	00	ABAR(I,K)=TIME(I,J)+XDIFF(I)	469
3		ABAR(I.NK(I))=TIME(I, 1)+XDIFF(I)	470
		IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 88	471
		NK1=NK(1)	472
		WRITE(6.227)NUMS(1).(ABAR(1.J).J=1.NK1)	473
	88	CONTINUE	474
		IF (LAMBDA.LT.LMIN) GO TO 998	475
		GC TO 99	476
C			477
C		UPCATE THE FLOW AFTER BREAKTHROUGH.	478
C			479
	21	IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 34	480
		WRITE(6,205)	481
	34	FLOW(I.KI(ITERM))=FLOW(I.KI(ITERM))+CAP(SINK)	482
c		IF DIREC =0 THEN CAP ADDED TO FLOW.	483
C		IF DIREC =1 THEN CAP IS SUBTRACTED.	484
	30	ITERM=CRIG2(ITERM)	485
c		CHECK IF BACK AT SOURCE.	486
		IF(ITERM.EQ.SOURCE) GO TO 33	487

		I=DIREC(ITERM)	488
		I=IABS(I)	489
		IF (DIREC(ITERM).GT.0) GC TO 34	490
		FLOW(I,KI(ITERM))=FLOW(I,KI(ITERM))-CAP(SINK) GC TO 30	491
c		RELABEL AND START OVER.	493
•	11	IF(TEST2.EQ.1.0K.TEST3.GE.1) GO TO 415	494
	-	DO 560 I=1.NA	495
		NK1=NK(I)	496
		DO 560 K=1.NK1	497
56	0	bRITE(6,220) NUMS(1) . K. FLOW(1.K)	498
		DO 98 I=1,LNODEN	499
		LABEL(I)=0	500
	98	CONTINUE	501
		GO TO 99	502
C		PREGRAM TERMINATES WHEN EVENTUALLY AN INFINITE FLOW IS ACHIEVED	503
C		FROM THE SOURCE TO THE SINK. OR WHEN THE VALUE OF LAMBDA DROPS	504
C		BELOW THE MINIMUM LENGTH OF THE NETWORK.	505
	998	IF(TEST3.NE.0) GU TO 999	506
		BRITE(6.202) LAMBDA	507
		60 10 999	508
	705	WRITE(6.233) LINPUT.LMIN	509
		GO TO 999	510
	704	WRITE(6.236) LINPUT.LMAX	511
		WRITE(6.238) LINPUT	512
		D=0	513
		DC 60 I=1,NA	514
	60	IP(1)=0	515
		GO TO 707	516
		WRITE(6.234) LINPUT	517
	706	WRITE(6.238) LINPUT	518
		D=LINFUT-LAMBDA	519
	707	PCOST=0.	520
		DO 57 I=1.NA	521
		IF(IP(I).EQ.1.AND.D.GT.0) XACT(I)=XACT(I)-D	522
		IF(IP(I).EQ.2.AND.D.GT.0) XACT(I)=XACT(I)+D	523
		PCOST1=0.	524
		NKM1=NK(1)-1	525
		00 51 K=1, NKM1	526
		IF(K.NE.1) GO TO 52 XIJ=XACT(1)	527 528
		IF(XIJ-GT-TIME(1-2)) XIJ=TIME(1-2)	529
		GO TO 53	530
	52	XIJ=XACT(I)-TIME(I.K)	531
	32	IF(XIJ-LT-0) XIJ=0	532
		IF(XIJ.GT.(TIME(I.K+1)-TIME(I.K))) XIJ=TIME(I.K+1)-TIME(I.K)	533
	53	PCOSTI=PCOSTI+C(I.K)*XIJ	534
		CONTINUE	535
	٠.	KCOST=COST([,1]+C([,1]+T[ME([,1])	536
		ACOST=KCOST-PCOST1	537
		BRITE(6.216)NUMS(1).XACT(1).ACDST	538
	57	PCGST=PCGST+ACGST	539
		WRITE(6.239) PCOST	540
	999	WRITE(6.228)	541
		READ(FO)	542
		WRITE(FO)KKKK,(FILEO(I), I=1,KKKK)	543
		ENOFILE F6	544
		ENCFILE FO	545
		WRITE(6.228)	546
		STOP	547
	100	FORMAT(14.1X.14.1X.11.1X.11.1X.14.1X.14.1X.14.1X.11)	548

```
150 FORMAT( *- *, *THE NUMBER OF NODES IS *, 14. *. *, /, 1X. *THE NUMBER OF AC
                                                                             549
   ITIVITIES IS ". 14. " . " . / . 1x . "THE SOURCE NODE IS NUMBERED " . 14 . " AND
                                                                             550
   2THE SINK NODE IS NUMBERED .. I4. .... .. ** NCDES: ***)
                                                                             551
151 FORMAT('0',16X.'K',7X.9(3X,14.5X))
                                                                             552
152 FORMAT( *- *, * ** ACTIVITIES: *** .//.6x. 11 .7x. 'XACT' .6x. 'ORIG' . 3x.
   1*TERM*,4X,*J*,6X,*TIME*,9X,*COST*,14X,*C*,13X,*ABAR*)
                                                                             554
153 FORMAT( * .3x.14.3x.110.3x.14.3x.14.(139.12.3x.110.3x.110.3x.
                                                                             555
   1E16.5.3X.110))
                                                                             556
154 FORMAT( -- . . THE ENTIRE PROJECT COST CURVE WILL BE DETERMINED . . )
                                                                             557
155 FORMAT( -- . . THE OPTIMAL ACTIVITY COMPLETION TIMES FOR A SPECIFIED
                                                                             558
   1PROJECT CEADLINE TIME = ",110." ARE GOING TO BE DETERMINED.")
                                                                             559
156 FCFMAT( * . 139.12.3x.110.3x.110.22x.110)
                                                                             560
157 FORMAT ( 0 ..
                                         4x. 'INITIAL XNODE(K)'.3X.
                                                                             561
   19(110.2X))
                                                                             562
200 FORMAT( *0 *, *LAMBDA = PROJECT COMPLETION TIME *, //,
                                                                             563
            1X. THE STARTING VALUE OF LAMBOA IS ......
  1
                                                                             564
            .1x. THE CORRESPONDING TOTAL PROJECT COST IS .E16.5. ..)
                                                                             565
566
202 FORMAT('0',///,30x,'* * * * *,////,1X,
                                                                             567
               THE SINK WAS REACHED WITH INFINITE CAPACITY IMPLYING A
                                                                             568
   IN INFEASIBLE SOLUTION TO THE PRIMAL PROBLEM .... 20x. IF LAMBDA DRO
                                                                             569
  2PS BELOW ITS CURRENT VALUE. '. [10. ...)
203 FORMAT( "-", "THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY -
                                                                             571
   1 CONTINUE WITH THE LABELING PROCESS. . . . 1X. THE NODES THAT HAVE
                                                                             572
   28EEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERA
                                                                             573
  STICN. 1)
                                                                             574
204 FORMAT( *0 , *THE NODE *. 14. * HAS THE LABEL ( *. 14. *. *. 14. *. *. 14. *. *. *.
   1E16.5. 1. )
                                                                             576
205 FGRMAT( *- *, *BREAKTHROUGH: UPDATE THE DUAL VARIABLES. *
                                                                             577
   1.///.1X. ACTIVITY #: 1'.3X.'J'.9X, 'NEW FLOW: F(1.J)')
                                                                             578
206 FDEMAT( "- ", "NUMBREAKTHROUGH: UPDATE THE PRIMAL VARIABLES: " . / . 1X .
                                                                             579
   1º1.E. DETERMINE OPTIMAL ACTIVITY TIMES FOR LAMBDA = ".110. ...)
                                                                             580
207 FORMAT( * .. NODE #: K .. 5X, 'NEW VALUE: XNODE(K) )
                                                                             581
209 FCRMAT( "- " DELTA (REPRESENTED BY "D") RANGES FROM O TO"
                                                                             582
   583
                       /.IX. THE MINIMUM COST PROJECT SCHEDULE FOR PRO
                                                                             584
   3JECT DEACLINE = ". 110. '-D:")
                                                                             585
210 FORMAT( .. 7x. 14.12x.110)
                                                                             586
211 FORMAT(' '.7X.14.12X.110. -D')
                                                                             587
212 FCFMAT( *- *, *PROJECT COMPLETION TIME = *. IIO. *-D. *,//. 1X.
                                                                             SAR
               ACTIVITY #: I'.3X, NEW VALUE: XACT(I)'.9X, ACTIVITY CO
                                                                             589
  25T1)
                                                                             590
213 FORMAT( * *.5x.14.12x.110. -D*.9x.E16.5. + ( *.E13.5. **D) *)
                                                                             591
214 FORMAT(' '.5x,14,12x,110,'+D',9x,E16.5,' + (',E13.5,'+D)')
216 FORMAT( * .5x.14.12X.110.11X.E16.5)
                                                                             593
220 FCRMAT( 1,12x,14,2x,12,7x,E16.5)
                                                                             594
224 FORMAT( *0 . . THE CURRENT VALUE OF THE PROJECT COST IS . . E16.5.
                                                                             595
   1' + ('.E13.5,'*D).')
                                                                             596
225 FORMAT( '-', ***
                       ITERATION NUMBER . 16.
                                                                             597
226 FORMAT( *- *, *NEW VALUES OF ABAR FOR J=1.2.... NK(I) *.//.6X.* I *.3X.
                                                                             598
   1'J:'.11(5x,12,3x))
                                                                             599
227 FORMAT( ',2x,14,7x,11(18,2X))
                                                                             600
228 FORMAT (1H1)
                                                                             601
230 FCFMAT(14.1X.14.1X.12)
                                                                             602
231 FLRMAT (8110)
                                                                            603
232 FORMAT(110)
                                                                             604
233 FORMAT ( '- ', 'THE SPECIFIED VALUE OF LAMBDA . ', IIO . ', IS LESS THAN THE
                                                                            605
   I MINIMUM VALUE. .. 110, .. IMPLYING AN INFEASIBLE SOLUTION .. .//. IX.
                                                                             606
  2'THE PROBLEM WILL NOT BE WORKED. ')
                                                                             607
234 FORMAT( 11. THE SPECIFIED VALUE OF LAMBDA. . . 110. . HAS BEEN REACHED
                                                                            608
   1.")
                                                                             609
```

```
235 FCFMAT ('0'.
                   THE SOURCE HAS A VALUE OF ZERO AND IS ASSIGNED THE
                                                                                  610
     3LABEL (-.-.- INF) . . . // )
                                                                                  611
  236 FORMAT( '-'. THE SPECIFIED VALUE OF LAMBDA, '. 110. . IS GREATER THAN
                                                                                  612
     I OR EQUAL TO THE MAXIMUM VALUE. .........
                                                                                  613
     1
                                              1X. THEREFORE. THE ORIGINAL
                                                                                  614
     2XNCDE(K) "S AND XACT(I) "S ARE OFTIMAL.")
                                                                                  615
  237 FORMAT( * * * * * * WARNING: ACTIVITY NUMBER * . 14. * HAS A NON-CONVEX C
                                                                                  616
     LOST FUNCTION: "./.12x, "IE. THE C(". 14. ". M) "S ARE NOT NON-INCREASIN
                                                                                  617
     2G. 1)
                                                                                  618
  238 FORMAT( -- . . FOR PROJECT COMPLETION TIME = .. . . . . THE CPTIMAL SOL
                                                                                  619
     1UTION 15: .//.1X.
                                                                                  620
                * ACTIVITY #: I*.3x. NEW VALUE: XACT(I)*.9x. ACTIVITY CO
                                                                                  621
     2ST 1)
                                                                                  622
  239 FORMAT( '- '. 'THE CORRESPONDING PROJECT COST IS '. E16.5. '. ')
                                                                                  623
      END
                                                                                  624
      SUBROUTINE ORDER
                                                                                  625
C
                                                                                  626
         THIS SUBROUTINE DETERMINES THE ORDER IN WHICH TO CONSIDER
C
                                                                                  627
C
         THE ACTIVITIES FOR THE CALCULATION OF THE CRITICAL PATH TIME
C
                                                                                  629
      IPPLICIT INTEGER+2(A-Z)
                                                                                  630
      INTEGER#4 NA, NN, ORIG. TERM. NK . LNODEN
                                                                                  631
      CCMMCN OFIG.TERM.NK.NA.LNCDEN.TIME.CTIME.XNODE.ACRD.LMIN.LMAX.
                                                                                  632
             TESTI
                                                                                  633
      CIMENSION ORIG(1000). TERM(1000). AORD(1000). CTIME(1000).
                                                                                  634
     **NCDE(9999).ND(9999).NDD(9999).TIME(1000.6).NK(1000)
                                                                                  635
      N=LNGDEN
                                                                                  636
      MENA
                                                                                  637
      NDD(1)=1
                                                                                  638
      DC 5 1=2.N
                                                                                  639
      NDD(1)=0
                                                                                  640
      DC 6 1=1.M
                                                                                  641
      ACRD(I)=0
6
                                                                                  642
      K=0
                                                                                  643
      MP=M+1
                                                                                  644
      DO 1 11=1.MP
                                                                                  645
      DC 20 I=1.N
                                                                                  646
20
      ND(I)=NDD(I)
                                                                                  647
      III=0
                                                                                  648
      IP=11+1
                                                                                  649
      DO 2 J=1.M
                                                                                  650
      IF (ND(ORIG(J)).NE.II) GO TO 2
                                                                                  651
      NDD(TERM(J))=IP
                                                                                  652
      111=1
                                                                                  653
      IF (K.EC.O) GO TO 14
                                                                                  654
      DO 10 L=1.K
                                                                                  655
      IF (AORG(L).EQ.J) GO TO 11
                                                                                  656
10
      CONTINUE
                                                                                  657
                                                                                  658
14
      K=K+1
      GO TO 13
                                                                                  659
11
      IF(L.EG.K) GU TO 2
                                                                                  660
      KM=K-1
                                                                                  661
      00 12 LL=L.KM
                                                                                  662
12
      ACRD(LL)=ACRD(LL+1)
                                                                                  663
13
      AORD (K)=J
                                                                                  664
      CENTINUE
                                                                                  665
2
      IF(III.EQ.0) GO TO 3
                                                                                  666
      CONTINUE
1
                                                                                  667
                                                                                 668
3
      CONTINUE
      DO 30 1=1.NA
                                                                                  669
30
      CTIME(1)=TIME(1.1)
                                                                                  670
```

AD-A060 568

TEXAS A AND M UNIV COLLEGE STATIST OF STATISTICS

A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJECT—ETC(U)

NOUNCLASSIFIED

THEMIS-TR-57

AUG 78

AD-A060 568

TEXAS A AND M UNIV COLLEGE STATIST OF STATISTICS

F/6 9/2

A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJECT—ETC(U)

NOUNCLASSIFIED

AUG 78

AUG 78

AD-A060 568

TEXAS A AND M UNIV COLLEGE STATIST OF STATISTICS

F/6 9/2

A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJECT—ETC(U)

NOUNCLASSIFIED

AUG 78

	LMIN=CPTIME(CPATHT)	671
	NK1=NK(1)	673
31		674
-	LMAX=CPTIME(I,NKI)	675
	RETURN	676
	ENG	677
	FUNCTION CPTIME(CPAINT)	678
c		679
č	DETERMINE THE CRITICAL PATH TIME: CPTIME	680
č	XNODE(I) = EARLIEST TIME THAT AN ACTIVITY BEGINNING AT NODE I	681
c	CAN COMMENCE	682
c	AND THE RESERVE OF THE PROPERTY OF THE PROPERT	683
•	IMPLICIT INTEGER+2(A-Z)	684
	INTEGER*4 NA.NN.ORIG.TERM.NK.LNODEN	685
	COMMON ORIGITERMINKINALLNODENITIMEISTIMEIXNODEIACROLLMINILMAXI	686
	* TEST1	687
	DIMENSION ORIG(1000).TERM(1000).ADRD(1000).CTIME(1000).	688
	*XNCDE(9999).ND(9999).NDD(9999).TIME(1000. 6).NK(1000)	689
	00 1 I=1.LNODEN	650
1	XNQCE(1)=0	691
•	DO 2 11=1.NA	692
	1=AORD(11)	693
2	IF(XNODE(DRIG(1))+CTIME(1).GT.XNODE(TERM(1)))	694
-	1 XNODE(TERM(1))=XNCDE(ORIG(1))+CTIME(1)	695
	CPTIME=XNGDE(TERM(11))	696
	II=ACRD(NA)	657
	RETURN	698

3. DSR

```
C
      DETERMINISTIC SCHEDULE RESOLUTION PROGRAM
C
C
         THIS PROGRAM ASCERTAINS THE DISTRIBUTIONAL PARAMETERS FOR EACH
         ACTIVITY IN THE PROJECT NETWORK THAT CORRESPOND TO THE MINIMUM
C
C
         COST PROJECT SCHEDULE WHICH YIELDS A DETERMINISTIC COMPLETION
                       THIS IS DONE BY REFERRING TO THE OUTPUT FROM THE
C
         TIME OF PD.
                                                                                   6
         DETERMINISTIC PROJECT SCHEDULER WHICH HAS BEEN STORED ON DISK.
C
C
         THE ACTIVITY DISTRIBUTIONS ARE USED BY SIMPLIFICATION AND
C
                                                                                   9
C
         INDIRECTLY BY SUBNETWORK ANALYSIS.
                                                                                  10
C
                                                                                  11
         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET
C
                                                                                  12
         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL
                                                                                  13
                      PROJECT NETWORK
C
                                                                                  14
         BKMAX = THE MAXIMUM NUMBER CF BREAK POINTS ALLOWED IN THE
C
                                                                                  15
                      ENTIRE PROJECT'S TIME-COST CURVE
C
                                                                                  16
         NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR
                                                                                  17
C
C
                     EACH ACTIVITY
                                                                                  18
         CURRENTLY. MMAX=1000; BKMAX=3000; NCTMAX=6
C
                                                                                  19
C
      INTEGER+2 BREAK (BKMAX) . FILE6 (2+MMAX) . TYME (NCTMAX)
C
                                                                                  21
C
      DIMENSION TIME(4).NCT(MMAX).FILEO(4*MMAX+5).NUMS(MMAX)
                                                                                  22
C
      EQUIVALENCE (FILEO(3*MMAX+6).NCT(1))
                                                                                  23
                                                                                  24
      INTEGER $2 NEREAK.BREAK(3000).FILE6(2000).TYME(6)
                                                                                  25
      REAL *8 AA.XM. BB. ALPHA. BETA
                                                                                  26
      DIMENSION TIME(4).NCT(1000).FILE0(4005).NUMS(1000)
                                                                                  27
      EQUIVALENCE (FILEO(1).NA).(FILEO(6).NUMS(1)).(FILEO(3006).NCT(1))
                                                                                  28
      INTEGER FO/8/.F1/9/.F2/10/.
                                                                                  29
                                         F4/12/.
                                                        F6/14/
      INTEGER NORM(5)/'NURM'. 'AL'.3*' '/.BET(5)/'BETA'.4*' '/
                                                                                  30
      INTEGER RECT(5)/'RECT'.'ANGU'.'LAR(','UNIF'.'ORM)'/
                                                                                  31
      INTEGER FIXED(5)/'FIXE','D(CO','NSTA','NT)',' '/
                                                                                  32
C
                                                                                  33
      FCR EACH ACTIVITY--
                                                                                  34
C
     USES ORIGINAL TIME VECTOR AND RESULTS OF THE DETERMINISTIC PROJECT
C
                                                                                  35
     SCHEDULER TO CREATE A NEW TIME VECTOR. THIS DETERMINES THE ACTIV-
                                                                                  36
C
     ITY DISTRIBUTIONS TO BE USED BY SIMPLIFICATION AND INDIRECTLY BY
C
                                                                                  37
C
     SUBNETWORK ANALYSIS.
                                                                                  38
                                                                                  39
      WRITE(6.1500)
                                                                                  40
1500 FORMAT(1H1,132(***)/1X,132(***)/*OTHIS IS THE OUTPUT FROM THE DETE
                                                                                  41
     *RMINISTIC SCHEDULE RESOLUTION PROGRAM: DSR*
                                                                                  42
     *//1x.132(***)/1x.132(***))
                                                                                  43
      REWIND FO
                                                                                  44
      RESINC FI
                                                                                  45
      REWIND F2
                                                                                  46
      REWINC F4
                                                                                  47
      REMIND F6
                                                                                  48
                                                                                  49
      READ(FO)FILEO
      READ(F1)
                                                                                  50
      READ (F4) NCYC . PD . NFLAG
      BRITE(6,2000)PD
                                                                                  52
2000 FCRMAT(1HO./'OTHE MINIMUM COST PROJECT SCHEDULE FOR A TARGET TIME
                                                                                  53
     OCF '.F12.5.' IS AS FOLLOWS: '//'OACTIVITY: '.8X.
                                                                       . MEA
                                                                                  54
     +N'.8X. 'DISTRIBUTION'.14X. 'PARAMETERS'/)
                                                                                  55
      READ(FO)
                                                                                  56
      READ (FO) NBREAK . (BREAK (I) . I=1 . NBREAK)
                                                                                  57
      RENIND F6
                                                                                  58
      DO 1 I=1. NBREAK
                                                                                  59
      J=I
```

```
IF (PD.GE.BREAK(J))GO TO 2
                                                                                  61
1
      CONTINUE
                                                                                  62
      NEB= PREAK (NUREAK )
                                                                                  63
      WRITE(6.100)NBB.PD
                                                                                  64
      FCRMAT(1HO. THE SMALLEST POSSIBLE PROJECT COMPLETION TIME BASED ON
100
                                                                                  65
     * NON-RANDOM INDIVIDUAL ACTIVITY COMPLETION TIMES IS *.110, *.*/
                                                                                  66
     **OTHE UPTIMAL ACTIVITY COMPLETION TIMES FOR A TARGET TIM*.
                                                                                  67
     *E UF '.E14.5.' HAVE BEEN REQUESTED.')
                                                                                  68
      IF(NFLAG.EQ.1)GO TO 300
                                                                                  69
                                                                                  70
      WRITE(6.101)PD
                                                                                  71
      FORMAT( THE OPTIMAL ACTIVITY COMPLETION TIMES FOR A TARGET .
101
                                                                                  72
     * * TIME OF . E14.5. * WILL BE SUPPLIED. *)
                                                                                  73
                                                                                  74
      NFLAG=1
      REBIND F4
                                                                                  75
      WRITE(F4)NCYC.PD.NFLAG
                                                                                  76
                                                                                  77
      D=0.0
      I=NBREAK-1
                                                                                  78
      GU TO 200
                                                                                  79
300
      WRITE(6.102)
      FCRMAT( OTHIS IS THE SECOND CONSECUTIVE REQUEST FOR AN INFEASIBLE
102
                                                                                  81
     *PROJECT DEAULINE TIME. */ A FATAL ERROR WILL BE COMMITTED SO THAT
                                                                                  82
     *ALL REMAINING JOB STEPS WILL BE SKIPPED. 1)
                                                                                  83
      CALL ERRSET(251.1.-1.1)
                                                                                  84
      XW=DSQRT (-1.DO)
                                                                                  85
      GC TC 400
                                                                                  86
2
      NFLAG=0
                                                                                  87
      C=0.00
                                                                                  88
      1=1-1
                                                                                  89
      IF(1.EQ.0)GO TO 4
                                                                                  90
      D=BREAK(1)-PD
                                                                                  91
200
      DG 3 J=1.1
                                                                                  92
3
      READ(F6)
                                                                                  93
      N1=2+NA
                                                                                  94
      REAC(F6)(FILE6(1).1=1.N1)
                                                                                  95
      DO 700 JJ=1.NA
                                                                                  96
      J=JJ+2
                                                                                  97
      EMA=FILE6(J)+D+FILE6(J-1)
                                                                                  98
      NACT=NCT(JJ)
                                                                                  99
      ANCTI=NNCT-1
                                                                                 100
      READ(F1)(TYME(J).J=1.NNCT)
                                                                                 101
      DG 709 J=1.NNCT
                                                                                 102
      IF(EMA.GE.TYME(J))GO TO 709
                                                                                 103
      JK=J-1
                                                                                 104
      GO TC 708
                                                                                 105
709
      CONT INUE
                                                                                 106
      JK=NNCT1
                                                                                 107
708
      DO 707 J=1.NNCT1
                                                                                 108
      IF(J-JK)706,705,706
                                                                                 109
705
      READ(F1)IDIST.TIME
                                                                                 110
      GC TO 707
                                                                                 111
706
      READ(F1)
                                                                                 112
707
      CCNTINUE
                                                                                 113
      MFIX=ICIST+2
                                                                                 114
      GC TC (710.711.712.715). MFIX
                                                                                 115
710
      CCNTINUE
                                                                                 116
C
                                                                                 117
C
      BETA DISTRIBUTION
                                                                                 118
C
                                                                                 119
      IF(TIME(4) .GT.-1.DO)GO TO 7100
                                                                                 120
C
                                                                                 121
```

```
CONVERT THE MOST LIKELY PARAMETER. TIME(2). TO ALPHA AND BETA
                                                                                  122
                                                                                  123
      AA = TIME(1)
                                                                                  124
                                                                                  125
      XM=TIME(2)
      BB = TIME(3)
                                                                                  126
      CALL ALBET (AA.XM.BB. ALPHA.BETA)
                                                                                  127
      TIME(2)
                = ALPHA
                                                                                  128
      TIME (4)
                 = BETA
                                                                                  129
7100
      CONTINUE
                                                                                 130
           = TIME(2)
                                                                                  131
                         + 1.
      6
           = TIME(4)
                         + 1.
                                                                                 132
      XMIN = TIME(1)
                                                                                 133
      XMAX = TIME(3)
                                                                                  134
C
                                                                                 135
c
      COMPUTE NEW ALPHA, BETA, MIN TIME, AND MAX TIME
                                                                                  136
C
                                                                                 137
      UA = (XMAX-XMIN)*(A/(A+B)) + XMIN
                                                                                 138
      C = (UA-XMIN)/(XMAX-UA)
                                                                                 139
      V = A*B/((A+B)**2*(A+B+1.))
                                                                                 140
      TIME(1)
                 = XMIN+EMA/UA
                                                                                  141
                 = (C/(V*(C+1)*(C+1))-1.)/(C+1)-1.
      TIME(4)
                                                                                 142
      TIME(3)
                 = XMAX+EMA/UA
                                                                                 143
      TIME (2)
                 = C*(TIME(4)
                                 +1-1-1-
                                                                                 144
      WRITE(6,2101)NUMS(JJ), EMA, BET, TIME(1), TIME(E), TIME(2), TIME(4)
                                                                                 145
2101 FGRMAT(16.6X.F12.5.6X.5A4.6X.*MIN=*.F13.5.*. MAX=*.F15.5.*. ALPHA=
                                                                                  146
     ** .F12.5. . BETA= .F12.5)
                                                                                 147
      GC TC 713
                                                                                  148
711
      CONTINUE
                                                                                 149
                                                                                  150
      NORMAL DISTRIBUTION
C
                                                                                 151
                                                                                 152
      IF (TIME(4)
                     .EQ.1.) GO TO 714
                                                                                  153
                =(TIME(3) -TIME(1)
      TIME(3)
                                         1/6-
                                                                                 154
      TIME (4)
                 = 1.
                                                                                 155
      TIME(1)
                =0.
                                                                                 156
      TIME(3)
                 = EMA/TIME(2)
                                  *TIME(3)
                                                                                 157
      TIME (2)
                  = EMA
                                                                                 158
      WAITE(6.2201)NUMS(JJ).EMA.NORM.TIME(2).TIME(3)
                                                                                 159
2201
      FORMAT(16.6X.F12.5.6X.5A4.6X.*MEAN=*.F12.5.*. ST.DEV=*.F12.5)
                                                                                 160
      GO TO 713
                                                                                 161
712
      CCNTINUE
                                                                                 162
C
                                                                                 163
c
      UNIFORM DISTRIBUTION
                                                                                 164
                                                                                 165
                                            +TIME(3)
                              /(TIME(1)
      TDUM = 2. +EMA+TIME(1)
                                                                                 166
                 = 2. *EMA +TIME(3) /(TIME(1)
                                                 +TIME(3)
                                                                                 167
                 = TOUM
      TIME (1)
                                                                                 168
      WRITE(6.2301)NUMS(JJ), EMA, RECT, TIME(1), TIME(3)
                                                                                 169
      FCRMAT(16.6X.F12.5.6X.5A4.6X. MIN= +.F13.5. +. MAX= +.F15.5)
2301
                                                                                 170
      GO TO 713
                                                                                 171
715
      WRITE(6,2401)NUMS(JJ),EMA,FIXED.TIME(1)
                                                                                 172
      FCRMAT(16.6x.F12.5.6x.5A4.6x.'TIME='.F12.5)
                                                                                 173
2401
                                                                                 174
      CCNTINUE
      WRITE (F2) IDIST . TIME
                                                                                 175
700
      CONTINUE
                                                                                 176
400
      STOP
                                                                                 177
                                                                                 178
      END
      SUBROUTINE ALBET (AA, XM, BB, ALPHA, BETA)
                                                                                 179
C SUBROLTINE TO CALCULATE THE ALPHA AND BETA VALUES OF THE BETA DENSITY.
                                                                                 180
C N#6.SEE PAPER BY A.W. WUNTHAM AND R.E. COLE. INDUSTRIAL ENGINEERING DEPT.
                                                                                 181
                                                                                 182
```

```
GIVEN AA, BB, AND XM THE VALUES OF ALPHA AND BETA ARE DETERMINED
C
      FROM A AND THE STANDARD APPROXIMATION B.
                                                                              184
                                                                              185
                                                                              186
C
      CALLING ARGUMENTS.
C
                                                                              187
          1. AA IS THE MINIMUM TIME ESTIMATE.
                                                                              IAA
C
          2. XM IS THE MOST LIKLEY TIME ESTIMATE.
C
                                                                              189
C
          3. 88 IS THE MAXIMUM TIME ESTIMATE.
                                                                              190
          4. ALPHA AND BETA ARE THE VALUES CALCULATED
                                                                             191
C TERPOL IS A ARITHMETIC STATEMENT FUNCTION TO DO LINEAR INTERPOLATION.
                                                                              192
C IF X IS THE VALUE REQUIRED WHICH LIES BETWEEN THE VALUES OF X1 AND X2
                                                                              193
C IN A TABLEX X2 IS THE LARGER VALUES Y1 AND Y2 THE CORRESPONDING VALUES
                                                                              194
C IN THE TABLE.XYZ GREATER THAN Y GREATER THAN YIC.
                                                                              195
      IMPLICIT REAL+8(A-H.O-Z)
                                                                              196
      TERPOL(X1,X2,Y,Y1,Y2)=X1+((X2-X1)*(Y-Y1)/(Y2-Y1))
                                                                              197
      KK=-1
                                                                             198
                                                                              199
      K=0
      KA=AA+10-D0 + 0-1D0
                                                                             200
      KM=XM+10.D0 + 0.100
                                                                             201
      K8=88+10.00 + 0.100
                                                                             202
      KH=KH-KM
                                                                             203
      KA=KM-KA
                                                                             204
      IF(XM.EQ.BB) GOTO 22
                                                                             205
      IF (XM.EQ.AA) GOTO 21
                                                                             206
      GOTO 23
                                                                             207
   21 ALPHA=0.00
                                                                             208
      BETA=2.872400
                                                                             209
      GCTO 20
                                                                             210
   22 ALPHA=2.872400
                                                                             211
      BETA=0.DO
                                                                             212
      GCTG 20
                                                                             213
   23 IF (KA-KB) 5.7.5
                                                                             214
     S= (XM-AA)/(BB-XM)
                                                                             215
      IF(S-1.DO) 6.7.8
                                                                             216
    7 ALPHA=3.00
                                                                             217
      BETA=3.00
                                                                             218
      GO TO 20
                                                                             219
    6 S=1.00/S
                                                                             220
      K=-1
                                                                             221
    8 Y=6.00
                                                                             222
      DC 10 1=1.600
                                                                             223
      Y=Y-1.D-2
                                                                             224
                                                                             225
      C=(Y**3)+(7.00*(Y**2))-20.00*Y -24.00
                                                                             226
      R=(B++2)-(144.D0+C)
                                                                             227
      FOOT=CSQRT(R)
                                                                             228
      ALF=(B+600T)/72.00
                                                                             229
      BET=(B-ROCT)/72.00
                                                                             230
      SS=ALF/BET
                                                                             231
      IF (SS- S) 10.11.14
                                                                             232
  14 IF (KK)15,16,11
                                                                             233
    KK=KK+1
      S1=SS
                                                                             235
      X1=ALF
                                                                             236
      Z1=BET
                                                                             237
      GOTO 10
                                                                             238
  10 CENTINUE
                                                                             239
      WRITE (6.18)
                                                                             240
  18 FURMAT(1H1. 32H ERROR.NO VALUE OF ALPHA BETA
                                                                             241
      CALL EXIT
                                                                             242
     KK=KK+1
                                                                             243
```

* '

	X2=ALF	244
	22=BET	245
	S2=SS	246
	ALF=TERPOL(X1.X2.S.S1.S2)	247
	BET=TERPOL(21,22,5,51,52)	248
11	IF (K) 12.13.13	249
12	BETA =ALF	250
	ALPHA = BET	251
	GC TO 20	252
13	ALPHA = ALF	253
	BETA = BET	254
20	RETURN	255
	END	256

```
4. SIMP
C
      SIMPLIFICATION PROGRAM
                                                                                   2
C
         THIS PROGRAM DETERMINES THE SIMPLIFIED PROJECT NETWORK BY
                                                                                   3
C
         REPLACING CERTAIN CONFIGURATIONS OF ACTIVITIES WITH SINGLE
C
                                                                                   4
C
         ACTIVITIES WHOSE DISTRIBUTIONS CAN BE DETERMINED TO MATCH THE
c
         DISTRIBUTIONS OF THE CONFIGURATIONS OF ACTIVITIES REPLACED.
                                                                                   6
C
                                                                                   7
         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS. LET
                                                                                   8
C
         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL
C
                                                                                   9
c
                     PROJECT NETWORK
                                                                                  10
C
         CURRENTLY. MMAX=1000
                                                                                  11
c
                                                                                  12
C
      INTEGER FILE3(3*NA) . S(NA.4)
                                                                                  13
•
                                                                                  14
      IMPLICIT REAL *8(A-H.O-R.T-Z)
                                                                                  15
      INTEGER FO/8/.
                                  F3/11/.
                                                                F7/15/
                                                                                  16
                        FILE3(3000).S(1000.4).LIST(24).JP(20)
                                                                                  17
      INTEGER
      EQUIVALENCE (FILE3(1).S(1))
                                                                                  18
      EQUIVALENCE (ISBR.LIST(1)).(1.LIST(2)).(JP(1).LIST(3))
                                                                                  19
      WRITE(6.9000)
                                                                                  20
9000 FORMAT(1H1.132(***)/1x.132(***)/*OTHIS IS THE OUTPUT FROM THE SIMP
                                                                                  21
     *LIFICATION PROGRAM: SIMP*/1H0.132(***)/1X.132(***))
                                                                                  22
      REWIND FO
                                                                                  23
      ARED=0
                                                                                  24
      READ (FO) NACT . NODES . NSRCE . NSI NK . L NODEN . S
                                                                                  25
      REAC(FO) JMAT
                                                                                  26
      NK=NACT
                                                                                  27
      REMIND F3
                                                                                  28
      READ(F3)
                                                                                  29
      DO 99 1=1.NK
                                                                                  30
99
      5(1.4)=1
                                                                                  31
                                                                                  32
C
                                                                                  33
         BUILD FILET FOR THE MODIFIED SIMPLIFICATION PROGRAM
C
                                                                                  34
C
                                                                                  35
C
      START OF SEARCH FOR REDUCIBLE NETWORKS
                                                                                  36
                                                                                  37
  100 CO 190 I=1.NK
                                                                                  AF
      IF(S(1.4).LE.0) GO TO 190
                                                                                  39
      KP=0
                                                                                  40
      CO 121 J=1.NK
                                                                                  41
      IF(1.NE.J.AND.S(J.4).GT.O.AND.S(1.2).EQ.S(J.2).AND.S(1.3).EQ.S(J.3
                                                                                  42
     1)) GG TG 122
                                                                                  43
      GC TG 121
                                                                                  44
  122 KP=KP+1
                                                                                  45
      JP (KP)=J
                                                                                  46
      IF(KP.GE.20) GO TO 200
                                                                                  47
  121 CONTINUE
                                                                                  48
      IF (KP) 120.120.200
                                                                                  49
  120 DO 130 J=1.NK
                                                                                  50
      IF(1.NE.J.AND.S(J.4).GT.O.AND.S([.3).EQ.S(J.2)) GO TO 125
                                                                                  51
  130 CENTINUE
                                                                                  52
      GO TO 190
                                                                                  53
  125 J1=J
                                                                                  54
      JK=J1+1
                                                                                  55
      LF (JK-NK) 127.127.141
                                                                                  56
  127 DC 140 J=JK.NK
                                                                                  57
      IF(1.NE.J.AND.S(J.4).GT.O.AND.S(J.2).EQ.S(J1.2)) GO TO 160
                                                                                  58
  140 CCNTINUE
                                                                                  59
  141 DG 150 J=1.NK
                                                                                  60
```

```
IF(1.NE.J.AND.J1.NE.J.AND.S(J.4).GT.O.AND.S(J.3).EQ.S(1.3)) GO TO
                                                                                   61
     1190
                                                                                   62
  150 CCATINUE
                                                                                   63
      GO TO 300
                                                                                   64
  160 J2=J
                                                                                   65
      K3=-1
                                                                                   66
      K4=-1
                                                                                   67
      K5=-1
                                                                                   68
  161 DO 165 J=1.NK
                                                                                   69
      IF(I.EQ.J.QR.JI.EQ.J.QR.J2.EQ.J.QR.S(J.4).LE.0) GO TO 165
                                                                                   70
      IF(S(J.3).EQ.S(1.3))GO TO 190
                                                                                   71
  165 CONTINUE
                                                                                   72
C
                                                                                   73
C
      CHECK FOR FIRST ORDER CROSSED NETWORKS
                                                                                   74
C
                                                                                   75
   49 KK=0
                                                                                   76
   50 DU 60 J=1. NK
                                                                                   77
      IF(1.EQ.J.OR.J.EQ.J1.OR.J.EQ.J2.OR.S(J.4).LE.0) GO TO 60
                                                                                   78
      IF(S(J.2).EQ.S(1.2).AND.S(J.3).EQ.S(J2.3)) GC TO 51
                                                                                   79
      GO TO 60
                                                                                   80
   £1 J3=J
                                                                                   81
   52 K3=K3+1
                                                                                   82
      IF(K3) 60.60.190
                                                                                   83
   60 CONTINUE
                                                                                   84
      IF (K3) 61.62.190
                                                                                   85
   61 IF(KK) 53.53.62
                                                                                   86
   53 L=J1
                                                                                   87
      J1=12
                                                                                   88
      J2=L
                                                                                   89
      KK=1
                                                                                   90
      GC TC 50
                                                                                   91
   62 DO 170 J=1.NK
                                                                                   92
      IF(1.EG.J.UR.J.EQ.J1.OR.J.EQ.J2.OR.S(J.4).LE.0) GO TO 170
                                                                                   93
      IF(S(J.2).EQ.S(J2.3).AND.S(J.3).EQ.S(J1.3)) GC TC 167
                                                                                   94
      IF(S(J,2).EQ.S(J1.3).AND.S(J.3).EQ.S(J2.3)) GO TO 168
                                                                                   95
      GO TO 170
                                                                                   96
  167 J4=J
                                                                                   97
      K4=K4+1
                                                                                   98
      IF(K4)170.170.190
                                                                                   99
  168 J5=J
                                                                                  100
      K5=K5+1
                                                                                  101
      1F(K5)170.170.190
                                                                                  102
  170 CONTINUE
                                                                                  103
      IF(K3.EQ.O.AND.K4.EQ.O.AND.K5.EQ.-1)GO TO 180
                                                                                  104
      IF(K3.EQ.O.AND.K4.EQ.-1.AND.K5.EQ.O) GO TO 171
                                                                                  105
      GO TO 190
                                                                                  106
  171 K6=-1
                                                                                  107
      K7=-1
                                                                                  108
      DC 175 J=1.NK
                                                                                  109
      IF(J.EG.1.OR.J.EG.J1.OR.J.EG.J3.OR.J.EG.J5.OR.S(J.4).LE.O.OR.J.EG.
                                                                                  110
     1J2) GO TO 175
                                                                                  111
      IF(S(J.2).EQ.S(J1.3)) GO TO 172
                                                                                  112
      IF(S(J.2).E0.S(J3.3)) GG TO 173
                                                                                  113
      IF(S(J.3).EQ.S(J1.3).OR.S(J.3).EQ.S(J3.3).OR.S(J.2).EQ.S([.3)) GO
                                                                                  114
     1TC 150
                                                                                  115
      GC TO 175
                                                                                  116
  172 J6=J
                                                                                  117
      K6=K6+1
                                                                                  118
      1F(K6)175.175.190
                                                                                  119
  173 J7=J
                                                                                  120
      K7=K7+1
                                                                                  121
```

```
IF (K7) 175.175.190
                                                                                 122
  175 CONTINUE
                                                                                 123
      IF (K6.EQ.O.AND.K7.EQ.O) GO TO 176
                                                                                 124
      GO TO 190
                                                                                 125
  176 IF(S(J6.3).EQ.S(J7.3)) GO TO 600
                                                                                 126
      GC TC 190
                                                                                 127
  180 K8=-1
                                                                                 128
      K9=-1
                                                                                 129
      K12=-1
                                                                                 130
      K13=-1
                                                                                 131
      DO 185 J=1.NK
                                                                                 132
      IF(J.EQ.I.OR.J.EQ.JI.OR.J.EQ.J2.OR.J.EQ.J3.OR.J.EQ.J4.OR.S(J.4).LE
                                                                                 133
     1.0) GC TC 185
                                                                                 134
                                                                                 135
      IF(S(J.2).Eu.S(J3.3)) GO TO 181
      IF(S(J.3).EQ.S(J3.3)) GC TO 182
                                                                                 136
      IF(S(J.2).EG.S(J4.3)) GO TO 183
                                                                                 137
      IF(5(J.2).EQ.S(1.3)) GO TC 184
                                                                                 138
      GO TO 185
                                                                                 139
  181 J8=J
                                                                                 140
      K8=K8+1
                                                                                 141
      IF (K8) 185.185.190
                                                                                 142
  182 IF(S(J.2).EQ.S(J2.2)) GO TO 190
                                                                                 143
      19=J
                                                                                 144
      K9=K9+1
                                                                                 145
      IF (K9) 185,185,190
                                                                                 146
  183 J12=J
                                                                                 147
      K12=K12+1
                                                                                 148
      GC TU 185
                                                                                 149
  184 IF(S(J.3).EG.S(J1.3)) GO TO 190
                                                                                 150
      J13=J
                                                                                 151
                                                                                 152
      K13=K13+1
      IF(K13)185,185,190
                                                                                 153
  185 CCNTINUE
                                                                                 154
      IF(K13.EG.0.AND.K12.EQ.0.AND.K8.EQ.-1.AND.K9.EQ.-1) GO TO 186
                                                                                 155
      IF (K13.EQ.-1.AND.K12.EQ.O.AND.K8.EQ.O.AND.K9.EO.-1) GO TO 187
                                                                                 156
      IF(K13.EQ.-1.AND.K9.EQ.O.AND.K8.EQ.-1) GO TO 188
                                                                                 157
      IF(K13.EQ.-1.AND.K9.EQ.-1.ANC.K8.EQ.O) GO TO 189
                                                                                 158
      IF (K13.EQ.-1.AND.K9.EQ.-1.AND.K8.EQ.-1) GO TO 900
                                                                                 159
      GC TO 190
                                                                                 160
C
                                                                                 161
      CRISS CROSS 1
C
                                                                                 162
                                                                                 163
  186 IF(S(J13.3).NE.S(J12.3)) GO TC 190
                                                                                 164
      00 192 J=1.NK
                                                                                 165
      IF (J.EQ.I.OK.J.EQ.J1.OR.J.EQ.J2.OR.J.EQ.J3.OR.J.EQ.J4.OR.J.EQ.J12.
                                                                                 166
     1CR.J.EG.J13.OR.S(J.4).LE.0) GO TO 192
                                                                                 167
      IF(S(J.3).EQ.S(J4.3)) GC TO 190
                                                                                 168
  192 CENTINUE
                                                                                 169
       GC TG 400
                                                                                 170
C
                                                                                 171
C
      CRISS CRCSS 2
                                                                                 172
                                                                                 173
  187 IF(S(J8.3).NE.S(J12.3)) GO TO 190
                                                                                 174
       DO 193 J=1.NK
                                                                                 175
       IF(J.EQ.1.0R.J.EQ.J1.0R.J.EQ.J2.0R.J.EQ.J3.CR.J.EQ.J4.0R.J.EQ.J8.
                                                                                 176
     ICR.J.EG.J9.OR.S(J.4).LE.0) GO TO 193
                                                                                 177
      IF(S(J.3).EQ.S(J4.3)) GO TO 190
                                                                                 178
  193 CCATINUE
                                                                                 179
      GC TC 500
                                                                                 180
                                                                                 181
      CCUBLE WHEATSTONE BRIDGE 1
C
                                                                                 182
```

```
c
                                                                                  183
  188 K10=-1
                                                                                  184
      K11=-1
                                                                                  185
      DC 196 J=1.NK
                                                                                  186
      IF(J.EC.I.OR.J.EQ.JI.OR.J.EQ.J2.OR.J.EQ.J3.OR.J.EQ.J4.OR.J.EQ.J9)
                                                                                  187
     1GO TO 196
                                                                                  188
      IF(S(J.3).EQ.S(J9.2)) GO TO 194
                                                                                  189
      IF(S(J.2).EQ.S(J9.2)) GO TO 195
                                                                                  190
      GC TO 196
                                                                                  191
  194 IF(S(J.2).NE.S(J3.2)) GO TG 190
                                                                                  192
      J10=J
                                                                                  193
      K10=K10+1
                                                                                  194
                                                                                  195
      IF(K10)196.196.190
  195 IF(S(J.3).NE.S(J4.3)) GO TO 190
                                                                                  196
      J1 1= J
                                                                                  197
      K11=K11+1
                                                                                  198
      IF (K11) 196.196.190
                                                                                  199
  196 CONTINUE
                                                                                  200
      IF(K10.EQ.O.AND.K11.EQ.O) GO TO 700
                                                                                  201
      GC TC 190
                                                                                  202
C
                                                                                  203
C
      DOUBLE WHEATSTONE BRIDGE 2
                                                                                  204
C
                                                                                  205
  189 K10=-1
                                                                                  206
      K11=-1
                                                                                  207
      DC 199 J=1.NK
                                                                                  208
      IF(J.EQ.1.OR.J.EQ.J1.CR.J.EQ.J2.QR.J.EQ.J3.QR.J.EQ.J4.QR.J.EQ.J8 .
                                                                                  209
     10R.S(J.4).LE.U) GQ TQ 199
                                                                                  210
      IF(S(J.3).EQ.S(J8.3)) GO TO 197
                                                                                  211
      IF(S(J.2).EQ.S(J8.3)) GO TO 198
                                                                                  212
      GC TO 159
                                                                                  213
  197 IF(S(J.2).NE.S(J3.21) GO TO 190
                                                                                  214
      J10=J
                                                                                  215
      K10=K10+1
                                                                                  216
      IF(K10)199.199.190
                                                                                  217
  198 IF(S(J.3).NE.S(J4.3)) GO TC 190
                                                                                  218
                                                                                  219
      J11=J
      K11=K11+1
                                                                                  220
      IF(K11)199.199.190
                                                                                  221
  199 CONTINUE
                                                                                  222
      IF (K10.EQ.O.AND.K11.EQ.O) GO TO 800
                                                                                  223
  190 CONTINUE
                                                                                  224
      IF (NEED) 1103.1103.91
                                                                                  225
   91 NRED = 0
                                                                                  226
      GC TO 100
                                                                                  227
                                                                                  228
C
   BEGINNING OF PARALLEL REDUCTION SECTION
                                                                                  229
                                                                                  230
  200 NREU=NRED+1
                                                                                  231
      J1=JP(1)
                                                                                  232
      DO 201 J=2.KP
                                                                                  233
      L=JP(J)
                                                                                  234
                                                                                  235
      S(L.4) =-1
201
      5(1.4)=-1
                                                                                  236
      158R=1
                                                                                  237
                                                                                  238
      L15T(23)=KP
      LIST(24)=J1
                                                                                  239
      BRITE(F7) LIST
                                                                                  240
      IF (JMAT) 1040.1040.120
                                                                                  241
 1040 LIM=KP-1
                                                                                  242
      00 202 K=1.KP
                                                                                  243
```

```
IF=JP(K)
                                                                                  244
  202 JP(K)=S(1P.1)
                                                                                  245
                                                                                  246
      WRITE(6.1024) S(1.1),(JP(K),K=1,LIM)
      WRITE(6.1023) JP(KP)
                                                                                  247
 1024 FURMAT (1HO. 10X11H ACTIVITIES 14,(10(2H, 14)/12X))
                                                                                  248
 1023 FORMAT(IH . 10X4HAND 14.32H HAVE BEEN COMBINED IN PARALLEL.
                                                                                  249
                                                                                  250
      GC TC 1290
C
                                                                                  251
c
      SERIES REDUCTION
                                                                                  252
                                                                                  253
c
                                                                                  254
  300 NED=NRED+1
      5(1.4)=-1
                                                                                  255
                                                                                  256
      ISER=2
      LIST(3)=J1
                                                                                  257
      WRITE(F7) LIST
                                                                                  258
      IF (JMAT) 1009.1009.120
                                                                                  259
 1009 WRITE(6.1025) S(1.1),S(J1.1)
                                                                                  260
 1025 FORMAT(1HO, 42X11H ACTIVITIES 14.4H AND 14.20H COMBINED IN SERIES.)
                                                                                  261
      GO TO 1290
                                                                                  262
C
                                                                                  263
C
      CRISS CROSS TYPE 1 REDUCTION
                                                                                  264
                                                                                  265
  400 NEED=NREC+1
                                                                                  266
      5(1.4)=-1
                                                                                  267
      5(12.4)=-1
                                                                                  268
      5(13.4)=-1
                                                                                  269
      5( 14 .4)=-1
                                                                                  270
      5(112.4)=-1
                                                                                  271
      5(113,4)=-1
                                                                                  272
      158R=3
                                                                                  273
      L15T(3)=J1
                                                                                  274
                                                                                  275
      LIST(4)=J2
      L1ST(5)=J3
                                                                                  276
      LIST(6)=J4
                                                                                  277
      LIST(7)=J12
                                                                                  278
      LIST(8)=J13
                                                                                  279
      WALTE(F7) LIST
                                                                                  280
      IF(JMAT) 1230.1230.190
                                                                                  281
 1230 WRITE(6,1430)S(1,1),S(J1,1),S(J2,1),S(J3,1),S(J4,1),S(J12,1),S(J13
                                                                                  282
     1.1)
                                                                                  283
 1430 FORMAT(1+0.25x11HACTIVITIES 14,2H, 14.2H, 14.2H, 14.2H, 14.2H, 14.
                                                                                  284
     1 6H. AND 14.44H HAVE BEEN CCMBINED AS A CRISS CROSS TYPE 1.)
                                                                                  285
      GO TO 1290
                                                                                  286
                                                                                  287
C
      CRISS CROSS TYPE 2 REDUCTION
                                                                                  288
c
                                                                                  289
  500 NRED=NRED+1
                                                                                  290
      511.4)=-1
                                                                                  291
      5(12.4)=-1
                                                                                  292
      5(13.4)=-1
                                                                                  203
      5(14,4)=-1
                                                                                  294
      S(J8.4)=-1
                                                                                  295
      5(112.4)=-1
                                                                                  296
      ISBR=4
                                                                                  297
      L157(3)=J1
                                                                                  298
      LIST(4)=J2
                                                                                  299
      LIST(5)=J3
                                                                                  300
      L151(6)=J4
                                                                                  301
      L151(7)=J8
                                                                                  302
      LIST(8)=J12
                                                                                  303
      BRITE(F7) LIST
                                                                                  304
```

11

```
IF (JMAT) 1240.1240.190
                                                                                  305
 1240 WRITE(6.1440)S(1.1).S(J1.1).S(J2.1).S(J3.1).S(J4.1).S(J8.1).S(J12.
                                                                                  306
     11)
                                                                                  307
 1440 FCRMAT(1H0.25X 11HACTIVITIES 14.2H. 14.2H. 14.2H. 14.2H. 14.2H. 14.2H. 14.2H.
                                                                                  306
     1. 6H. AND I4.44H HAVE BEEN COMBINED AS A CRISS CROSS TYPE 2.)
                                                                                  309
      GC TO 1290
                                                                                  310
C
                                                                                  311
c
      CRISS CROSS TYPE 3 REDUCTION
                                                                                  312
C
                                                                                  313
  600 NRED=NRED+1
                                                                                  314
      5(1.4)=-1
                                                                                  315
      5(12.4)=-1
                                                                                  316
      5(13.4)=-1
                                                                                  317
      5(15,4)=-1
                                                                                  319
      5(16.4)=-1
                                                                                  319
      S(J7.4)=-1
                                                                                  320
      I SBR=5
                                                                                  321
      LIST(3)=J1
                                                                                  322
      LIST(4)=J2
                                                                                  323
      LIST(51=J3
                                                                                  324
      LIST(6)=J5
                                                                                  325
      LIST(7)=J6
                                                                                  326
      L1ST(8)=J7
                                                                                  327
      WRITE(F7) LIST
                                                                                  328
      IF(JMAY) 1250.1250.190
                                                                                  329
 1250 WRITE(6.1450)S(1.1).S(J1.1).S(J2.1).S(J3.1).S(J5.1).S(J6.1).S(J7.1
                                                                                  330
     1)
                                                                                  331
 1450 FCRMAT(1H0.25X 11HACTIVITIES 14.2H. 14.2H. 14.2H. 14.2H. 14.2H. 14
                                                                                  332
     1. 6H. AND I4.44H HAVE BEEN COMBINED AS A CRISS CROSS TYPE 3.)
                                                                                  333
      GO TO 1290
                                                                                  334
c
                                                                                  335
c
      DOUBLE WHEATSTONE BRIDGE TYPE I REDUCTION
                                                                                  336
                                                                                  337
  700 NEED=NREC+1
                                                                                  338
      5(1.4)=-1
                                                                                  339
      5(12.4)=-1
                                                                                  340
      S(J3.4)=-1
                                                                                  341
      S(J4.4)=-1
                                                                                  342
      S(J9.4)=-1
                                                                                  343
      5(110.4)=-1
                                                                                  344
      5(111.4)=-1
                                                                                  345
      ISBR=6
                                                                                  346
      LIST(3)=J1
                                                                                  347
      L1ST(4)=J2
                                                                                  348
      LIST(5)=J3
                                                                                  349
      LIST(6)=J4
                                                                                  350
      LIST(7)=J9
                                                                                  351
      LIST(8)=J10
                                                                                  352
      LIST(S)=JII
                                                                                  353
      WRITE(F7) LIST
                                                                                  354
      IF(JMAT) 1210.1210.190
                                                                                  355
 1210 WRITE(6.1410)S(1.1).S(J1.1).S(J2.1).S(J3.1).S(J4.1).S(J9.1).S(J10.
                                                                                  356
                                                                                  357
 1410 FORMAT(1HO.10X 11HACTIVITIES 14.2H. 14.2H. 14.2H. 14.2H. 14.2H. 14.2H.
                                                                                  358
     1.2h. I4.6h. AND I4. 57H HAVE BEEN COMBINED AS A DOUBLE WHEATSTONE
                                                                                  359
     28RIDGE TYPE 1.)
                                                                                  360
      GC TC 1290
                                                                                  361
C
                                                                                  362
C
       DOUBLE WHEATSTONE BRIDGE TYPE 2 REDUCTION
                                                                                  363
                                                                                  364
  800 NED=NRED+1
                                                                                  365
```

6,0

```
366
      5(1.4)=-1
                                                                                367
      S(J2.4)=-1
      5(13.4)=-1
                                                                                368
      5(14.4)=-1
                                                                                369
      5(18.4)=-1
                                                                                370
      5(110.4)=-1
                                                                                371
      5(111.4)=-1
                                                                                372
      ISER=7
                                                                                373
      LIST(3) = J1
                                                                                374
                                                                                375
      LIST (4)=J2
      LIST(5)=J3
                                                                                376
      LIST(6)=J4
                                                                                377
      LIST(7)=JE
                                                                                378
      LIST(8)=J10
                                                                                379
      LIST(9)=J11
                                                                                380
      WRITE(F7) LIST
                                                                                381
      IF(JMAT) 1220,1220,190
                                                                                382
 1220 WRITE(6.1420)S(1.1).S(J1.1).S(J2.1).S(J3.1).S(J4.1).S(J8.1).S(J10.
                                                                                383
                                                                                384
     11).S(J11.1)
 1420 FCFMAT(1H0.10X 11HACTIVITIES 14.2H. 14.2H. 14.2H. 14.2H. 14.2H. 14
                                                                                385
     1.2H. 14.6H. AND 14. 57H HAVE BEEN COMBINED AS A DOUBLE WHEATSTONE
                                                                                386
     2BRIDGE TYPE 2.)
                                                                                387
      GC TO 1290
                                                                                388
C
                                                                                389
C
      WHEATSTONE BRIDGE REDUCTION
                                                                                390
C
                                                                                391
  900 IF (NRED.EQ.O) NRED=NRED+1
                                                                                392
      5(1.4)=-1
                                                                                393
      5(12.4)=-1
                                                                                394
      S(J3.4)=-1
                                                                                395
      5(14.4)=-1
                                                                                396
      ISER=8
                                                                                397
      LIST(3)=J1
                                                                                398
      LIST(4)=J2
                                                                                399
      LIST(5)=J3
                                                                                400
      L1ST(6)=J4
                                                                                401
      WRITE(F7) LIST
                                                                                402
      IF (JMAT) 1201.1201.190
                                                                                403
 1201 WRITE(6.1401) S(1.1).S(J1.1).S(J2.1).S(J3.1).S(J4.1)
                                                                                404
 1401 FURMAT(1H0.20X 11HACITIVITES 14,2H. 14.2H. 14.2H. 14.6H. AND 14. 4
                                                                                405
     13H HAVE BEEN COMBINED AS A WHEATSTONE BRIDGE.)
                                                                                406
                                                                                407
 1290 WRITE(6.1490) 5(J1.1).5(J1.2).5(J1.3)
                                                                                408
 1490 FORMAT (1H .41X. 9HACTIVITY .14.18H NOW HAS TAIL NO. 14,14H AND HEA
                                                                                409
     10 NO. .[4.1H.)
                                                                                410
      GC TO 190
                                                                                411
1103 WRITE(6,1027)
                                                                                412
1027 FORMAT( 11 . 10x . THE FOLLOWING ACTIVITIES CANNOT BE FURTHER COMBINE
                                                                                413
     *D: *)
                                                                                414
      WEITE(6.2001)
                                                                                415
2001 FCFMAT(///11x. ACTIVITY CRIGIN TERMINAL.
                                                                                416
     *5x, ACTIVITY NUMBER ASSIGNED BY DECOMP')
                                                                                417
      NACT=0
                                                                                418
      DO 1105 I=1.NK
                                                                                419
      IF(S(1.4).LT.0)GO TO 1105
                                                                                420
      NACT=NACT+1
                                                                                421
     ##ITE(6.1028)S(1.1).S(1.2).S(1.3).NACT
1104
                                                                                422
     FORMAT (13x,14,6x,15,7x,15,9x,15)
                                                                                423
1028
1105 CONTINUE
                                                                                424
      ENDFILE F7
                                                                                425
C
                                                                                426
```

c	BUILD FILES FOR DECOMPOSITION	427
•	NACT = 0	429
	LNGDEN = 0	430
c	ERODEN - V	431
č	COUNT ACTIVITIES AND DETERMINE LARGEST NODE NUMBER	432
č	DETERMINE L U AND P FOR SUBNETWORK ANALYSIS	433
c	. Determine E o Millo P Tok Society Tokin Miller	434
	DO 1153 J=1,NK	435
c		436
c	S(J,4) IS -1 IF J-TH ACTIVITY HAS BEEN ELIMINATED	437
	IF(S(J.4).EQ1)GO TO 1153	439
	NACT=NACT+1	440
	S(NACT+1)=NACT	441
	S(NACT.2)=S(J.2)	442
	S(NACT.3)=S(J.3)	443
	IF (S(NACT.3).GT.LNODEN) LNODEN = S(NACT.3)	444
1153	CONTINUE	445
C		446
C	FIND SOURCE AND SINK	447
C		448
	NSRCE = 0	449
	NSINK = 0	450
	DO 1154 J=1,NACT	451
	IFLG = 0	452
	JFLG = 0	453
	(IRG = S(J,2)	454
	ITER = S(J.3)	455
	DO 1156 K=1.NACT	456
	IF (IIRG.EQ.S(K.3)) IFLG=1	457
	IF (ITER.EQ.S(K.2)) JFLG=1	458
1156	CONTINUE	459
	IF (IFLG.EQ.O) NSRCE = S(J.2)	460
	IF (JFLG.EQ.U) NSINK = S(J.3)	461
	IF (NSRCE.NE.O.AND.NSINK.NE.O) GO TO 1157	462
1154		463
1157	CCNTINUE	464
	WRITE(F3)NACT.NODES.NSRCE.NSINK.LNODEN.FILE3	465
	STOP	466
	FND	467

5. MODSIMP

```
MODIFIED SIMPLIFICATION PROGRAM
C
c
         THIS PROGRAM DETERMINES THE DISTRIBUTION OF EACH OF THE
         ACTIVITIES IN THE SIMPLIFIED PROJECT NETWORK (DETERMINED BY THE
C
C
         SIMPLIFICATION PRUGRAM).
C
         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET
C
c
         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL
                      PROJECT NETWORK
C
                                                                                   9
C
         CURRENTLY. MMAX=1000
                                                                                  10
C
                                                                                  11
      DIMENSION S(MMAX.4).CDF(MMAX.12)
C
                                                                                  12
                                                                                  13
      IMPLICIT REAL+8(A-H.O-R.T-Z)
                                                                                  14
      INTEGER FO/8/.
                         F2/10/.
                                          F4/12/.
                                                               F7/15/
                                                                                  15
      COMMON /BLKA/S.CDF.A.DENS.TI
                                                                                  16
      INTEGER S
                                                                                  17
      DIMENSION S(1000.4).CDF(1000.12).A(12).DENS(400).T1(2)
                                                                                  18
      COMMON /BLKB/TIME.Y
                                                                                  19
      DIMENSION TIME(4),Y(12)
      REAL #4 TYME(4)
                                                                                  21
      DIMENSION
                              JP(20).LIST(24)
                                                                                  22
      EQUIVALENCE (ISBR.LIST(1)).(I.LIST(2)).(JP(1).LIST(3))
                                                                                  23
      WRITE(6.9000)
                                                                                  24
9000 FORMAT(1H1,132(***)/1X,132(***)/*OTHIS IS THE OUTPUT FROM THE MODI
                                                                                  25
     *FIED SIMPLIFICATION PROGRAM: MODSIMP*/1H0,132(***)/1x,132(***))
                                                                                  26
      REBING FO
                                                                                  27
      REWIND F2
                                                                                  28
      REWIND F7
                                                                                  29
      A(1)=0.0
                                                                                  30
      A(12)=1.00
                                                                                  31
      DO 82 J=2.11
                                                                                  32
   82 A(J)= DFLGAT(J-2)+0.100 + 0.0500
                                                                                  33
      1500=C
                                                                                  34
      J200=0
                                                                                  35
      READ (FO ) NACT . NODES . NSRCE . NSINK . LNODEN . S
                                                                                  36
      IFLOP = 0
                                                                                  37
    5 DO 79 1=1.NACT
                                                                                  38
      READ (F2) MM. TYME
      MM=MM+2
                                                                                  40
      TIME(1) = TYME(1)
                                                                                  41
      TIME(2) = TYME(2)
                                                                                  42
      TIME(3) = TYME(3)
                                                                                  43
      TIME(4) = TYME(4)
                                                                                  44
      CALL XDENS(MM.1.1500.J200)
                                                                                  45
      S(1.4)=MM
                                                                                  46
      00 42 J=1.12
                                                                                  47
      CCF(1.J)=Y(J)
                                                                                  48
   79 CONTINUE
                                                                                  49
      NK=NACT
                                                                                  50
100
      READ(F7, END=99) LIST
                                                                                  51
      GO TO (200.300.400.500.600.700.800.900).ISBR
                                                                                  52
                                                                                  53
200
      CCNTINLE
                                                                                  54
      MARALLEL REDUCTION
                                                                                  55
      KP = LIST(23)
                                                                                  56
      J1 = LIST(24)
                                                                                  57
      CALL PARRED(I.JP.KP.JI)
                                                                                  58
      GC TG 100
                                                                                  59
300
      CENTINUE
```

```
c
                                                                                    61
      SERIES REDUCTION
                                                                                    62
                                                                                   63
      JI = LIST(3)
                                                                                   64
      5(1.4)=-1
                                                                                   65
      CALL SERRED(1,J1)
                                                                                   66
      GG TO 100
                                                                                   67
400
      CENTINUE
                                                                                   68
                                                                                   69
C
      CRISS CHOSS TYPE 1 REDUCTION
                                                                                   70
C
                                                                                   71
      J1 = LIST(3)
                                                                                   72
      J2 = L15T(4)
                                                                                   73
      J3 = LIST(5)
                                                                                   74
      J4 = LIST(6)
                                                                                   75
      J12= LIST(7)
                                                                                   76
      J13= LIST(8)
                                                                                   77
      5(1.4)=-1
                                                                                   78
      5(12.4)=-1
                                                                                   79
      S(J3.4)=-1
                                                                                   80
      5(14.4)=-1
                                                                                   81
      5(112.4)=-1
                                                                                   2
      S(J13.4) =-1
                                                                                   83
      CALL CCIRED(1.J1.J2.J3.J4.J12.J13)
                                                                                   84
      GC TO 100
                                                                                   85
500
      CENTINUE
                                                                                   86
C
                                                                                   87
C
      CRISS CROSS TYPE 2 REDUCTION
                                                                                   88
C
                                                                                   89
      J1 = LIST(3)
                                                                                   90
      J2 = LIST(4)
                                                                                   91
      J3 = LIST(5)
                                                                                   92
      J4 = LIST(6)
                                                                                   93
      J8 = LIST(7)
                                                                                   94
      J12= L15T(8)
                                                                                   95
      5(1.4)=-1
                                                                                   96
      5(12.4)=-1
                                                                                   97
      5(13.4)=-1
                                                                                   98
      5(14.4)=-1
                                                                                   99
      5(18.4)=-1
                                                                                  100
      5(112.4)=-1
                                                                                  101
      CALL CC2RED(1.J1.J2.J3.J4.J8.J12)
                                                                                  102
      GO TO 100
                                                                                  103
600
      CCATINUE
                                                                                  104
                                                                                  105
C
      CRISS CROSS TYPE 3 REDUCTION
C
                                                                                  106
                                                                                  107
      J1 = L1ST(3)
                                                                                  108
      J2 = LIST(4)
                                                                                  109
      J3 = LIST(5)
                                                                                  110
      J5 = LIST(6)
                                                                                  111
      J6 = LIST(7)
                                                                                  112
      J7 = LIST(8)
                                                                                  113
      5(1.4)=-1
                                                                                  114
      5(12.4)=-1
                                                                                  115
      5(13.4)=-1
                                                                                  116
      5(15.4)=-1
                                                                                  117
      5(16.4)=-1
                                                                                  118
      S(J7.4)=-1
                                                                                  119
      CALL CC3RED(1.J1.J2.J3.J5.J6.J7)
                                                                                  120
      GO TO 100
                                                                                  121
```

```
CONTINUE
                                                                                 122
700
                                                                                  123
C
c
      DOUBLE WHEATSTONE BRIDGE TYPE 1 REDUCTION
                                                                                 124
C
                                                                                 125
      JI = LIST(3)
                                                                                  126
      J2 = LIST(4)
                                                                                  127
                                                                                  128
      J3 = LIST(5)
      J4 = LIST(6)
                                                                                  129
                                                                                 130
      J9 = LIST(7)
                                                                                 131
      J10= L15T(8)
      J11= LIST(9)
                                                                                  132
                                                                                  133
      5(1.4)=-1
      5(12,4)=-1
                                                                                  134
                                                                                 135
      5(13.4)=-1
                                                                                 136
      5(14.4)=-1
      S(J9.4)=-1
                                                                                 137
                                                                                  138
      5(110.4)=-1
      5(111.4)=-1
                                                                                  139
      CALL CWIRED(1, J1, J2, J3, J4, J9, J10, J11)
                                                                                 140
      GC TO 100
                                                                                 141
      CENTINUE
                                                                                 142
800
C
                                                                                 143
       DOUBLE WHEATSTONE BRIDGE TYPE 2 REDUCTION
C
                                                                                 144
C
                                                                                 145
      JI = LIST(3)
                                                                                  146
                                                                                 147
      J2 = LIST(4)
                                                                                  148
      J3 = LIST(5)
      J4 = LIST(6)
                                                                                  149
                                                                                 150
      J8 = LIST(7)
      J10= L151(8)
                                                                                  151
                                                                                 152
      J11= LIST(9)
                                                                                  153
      5(1.4)=-1
                                                                                 154
      5(12,4)=-1
      5(13.4)=-1
                                                                                 155
      5(14.4)=-1
                                                                                  156
                                                                                 157
      5(18.4)=-1
      5(110.4)=-1
                                                                                  158
                                                                                 159
      5(111.41=-1
      CALL CH2RED(1.J1.J2.J3.J4.J8.J10.J11)
                                                                                 160
      GO TO 100
                                                                                  161
900
      CONTINUE
                                                                                  162
C
                                                                                  163
C
      WHEATSTONE BRIDGE REDUCTION
                                                                                  164
                                                                                  165
C
      J1 = LIST(3)
                                                                                  166
      J2 = LIST(4)
                                                                                  167
      J3 = LIST(5)
                                                                                  168
      J4 = LIST(6)
                                                                                  169
      5(1.4)=-1
                                                                                  170
      S(J2.4)=-1
                                                                                  171
      5(13.4)=-1
                                                                                  172
      5(14.4)=-1
                                                                                 173
      CALL WBRED(1.J1.J2.J3.J4)
                                                                                  174
99
      CCNTINUE
                                                                                  175
      PRINT RESULTING DISTRIBUTIONS
                                                                                  176
      DO 1100 I=1.NACT
                                                                                  177
      IF (S(1.4).LT.0) GO TO 1100
                                                                                 178
      TIME(1)=0.00
                                                                                  179
      TIME(3)=0.00
                                                                                 180
      DC 270 K=2.11
                                                                                  181
      TIME(1)=TIME(1)+CDF(1.K)
                                                                                 182
```

```
270
      TIME(3)=TIME(3)+COF(1.K)++2
                                                                                  183
      TIME(1)=TIME(1) +. 100
                                                                                  184
      TIME(3)=DSQRT(DMAX1(0.00.TIME(3)+.100-TIME(1)++2))
                                                                                  185
      IFLCP = IFLOP + 1
                                                                                  186
      IF (MGD(IFLOP.3).EQ.0) WRITE(6.2001)
                                                                                  187
      FCRMAT(1H1)
                                                                                  188
      ## TE (6.1028) $ (1.1) . $ (1.2) . $ (1.3) . TIME(1) . TIME(3)
                                                                                  189
1028 FORMAT (///11x. ACTIVITY
                                  ORIGIN
                                              TERMINAL
                                                             MEAN
                                                                       STAND
                                                                                  190
     *ARD DEVIATION*/13x.14.6x.15.7x.15.6x.F6.1. 8x.F7.2//32x.*F(T)
                                                                                  191
     . T.)
                                                                                  192
      DO 1102 MP=1.12
                                                                                  193
      WRITE(6.6000) A(MP).CDF(1.MP)
                                                                                  194
6000 FCRMAT(31XF5.2.3H -- F10.4)
                                                                                  195
1102 CONTINUE
                                                                                  156
1100
      CCNTINUE
                                                                                  197
C
                                                                                  198
      SAVE THE SIMPLIFIED ACTIVITY DURATION DISTRIBUTIONS FOR SUBNETWORK
C
                                                                                  199
C
      ANALYSIS
                                                                                  200
C
                                                                                  201
      REWING F4
                                                                                  202
      READ (F4)
                                                                                  203
      NACT=0
                                                                                 204
      00 11002 J=1.NK
                                                                                  205
      IF(S(J.4).GE.O)NACT=NACT+1
                                                                                 206
11002 CONTINUE
                                                                                  207
      WRITE (F4) NACT
                                                                                  208
      DG 1153 J=1.NK
                                                                                 209
      IF(S(J,4).GT.0) WRITE(F4) (CDF(J,K).K=1.12)
                                                                                  210
      CONTINUE
                                                                                 211
      ENDFILE F4
                                                                                  212
C
      END OF MAIN PROGRAM
                                                                                 213
C
                                                                                 214
      STOP
                                                                                  215
      END
                                                                                 216
      SUBROUTINE PARRED(1.JP.KP.J1)
                                                                                  217
      IMPLICIT REAL+8(A-H.O-R.T-Z)
                                                                                 218
      COMMON /BLKA/S.CDF.A.DENS.TI
                                                                                 219
      INTEGER S
                                                                                  220
      DIMENSION S(1000.4).CDF(1000.12).A(12).DENS(400).TI(2)
                                                                                 221
      DIMENSION JP(20)
                                                                                 222
      T1(1)=CDF(1.1)
                                                                                 223
      T1(2)=CDF(1.12)
                                                                                 224
      J1=JP(1)
                                                                                 225
      CO 201 J=1.KP
                                                                                  226
      L=JP(J)
                                                                                 227
      IF(CCF(L.1).GT.TI(1)) TI(1)=CDF(L.1)
                                                                                 228
      IF(CCF(L.12).GT.T((2)) T1(2)=CDF(L.12)
                                                                                 229
      IF(L.EQ.J1) GO TO 201
                                                                                 230
      S(L.4)=-1
                                                                                  231
  201 CONTINUE
                                                                                 232
      5(1.4)=-1
                                                                                 233
  213 L=1
                                                                                 234
      KM=10
                                                                                 235
                                                                                 236
      K=1
      DELTAT=(T1(2)-T1(1))/10.000
                                                                                 237
      MP=0
                                                                                 238
      M= 1
                                                                                 239
  217 DO 220 J=2.10
                                                                                 240
      K=K+1
                                                                                 241
      T=T1(1)+DFLUAT(J-1)+DELTAT
                                                                                 242
      IF(T.GE.CDF(L.12)) GO TO 221
                                                                                 243
```

1 . At 1994

218	IF(T.GE.CDF(L.M).AND.T.LT.CDF(L.M+1)) GO TO M=M+1	219
	GC TC 218	
219	IF(M.NE.1) GJ TO260	
	TI=CDF(L .1)	
	T2=COF(L .2)	
	T3=CCF(L .3)	
	¥1=0.00	
	Y2=0.05D0	
	¥3=0.1500	
~ ~	60 10 261	
200	T1=CDF(L,M)	
	T2=CDF(L.M+1)	
	T3=CDF(L.M-1)	
	Y1=A(M)	
	Y2=A(M+1)	*
	Y3=A(M-1)	
261	DENS(K)=TERPOL(Y1.Y2.Y3.T.T1.T2.T3)	
	IF(DENS(K).LT.0.DO) DENS(K)=0.DO	
	IF(DENS(K).GE.1.DO) GO TO 221	
20	CONTINUE	
_	GO TC 231	
2.	K0=K	
21	DO 230 K=KO.KM	
20		
-	DENS(K)=1.D0	
31	MP=MP+1	
	IF(MP-GT-KP) GO TO 240	
	L=JP(MP)	
	K=KM	
	M=1	
	KM=KM+9	
	GC TO 217	
	CONTINUE	
	DO 251 M=2,10	
	00 250 J=1.KP	
	L=M+J+9	
5 ^		
	DENS(M) =DENS(M) +DENS(L)	
31	CONTINUE	
	DENS(1)=0.000	
	DENS(11)=1.000	
	CALL INVSTR(J1.DELTAT)	
	RETURN	
	END	
	SUBROUTINE SERRED(1.J1)	
	IMPLICIT REAL+B(A-H.O-R.T-Z)	
	COMMON /BLKA/S.CDF.A.DENS.TI	
	INTEGER S	
	DIMENSIGN S(1000.4).CDF(1000.12).A(12).DENS	S(A00) - T1(2)
	TI(1)=CDF(1.1)+CDF(J1.1)	
	TI(2)=COF(1,12)+COF(J1,12)	
	IF(CDF(J1.12).EQ.CDF(J1.11) GC TO 370	
	IF(CDF(J1.12).EQ.CDF(J1.1)) GC TO 370 IF(CDF(1.12).EQ.CDF(1 .1)) GO TO 376	
	IF(CDF(J1.12).EQ.CDF(J1.1)) GC TO 370 IF(CDF(1.12).EQ.CDF(1.1)) GO TO 376 DENS(1)=0.00	
	IF(CDF(J1.12).EQ.CDF(J1.1)) GC TO 370 IF(CDF(1.12).EQ.CDF(1 .1)) GO TO 376	
	IF(CDF(J1.12).EQ.CDF(J1.1)) GC TO 370 IF(CDF(1.12).EQ.CDF(1.1)) GO TO 376 DENS(1)=0.00	
	IF(CDF(J1.12).EQ.CDF(J1.1)) GC TG 370 IF(CDF(1.12).EQ.CDF(I .1)) GC TG 376 DENS(1)=0.00 DENS(11)=1.000	
	IF(CDF(J1.12).EQ.CDF(J1.1)) GC TG 370 IF(CDF(1.12).EQ.CDF(I .1)) GC TG 376 DENS(1)=0.00 DENS(11)=1.000 TEMPO=TI(1)	
	IF(CDF(J1.12).EQ.CDF(J1.1)) GC TG 370 IF(CDF(1.12).EQ.CDF(I .1)) GO TO 376 DENS(1)=0.00 DENS(11)=1.000 TEMPO=TI(1) DELTAT=(TI(2)-TI(1))/10.000	
	IF(CDF(J1.12).EQ.COF(J1.1)) GC TG 370 IF(CDF(1.12).EQ.CDF(I .1)) GO TO 376 DENS(1)=0.00 DENS(11)=1.000 TEMPO=TI(1) DELTAT=(TI(2)-TI(1))/10.000 DO 399 J=2.10 K1=1	
	IF(CDF(J1.12).EQ.COF(J1.1)) GC TG 370 IF(CDF(1.12).EQ.CDF(I .1)) GO TO 376 DENS(1)=0.00 DENS(11)=1.000 TEMPO=TI(1) DELTAT=(TI(2)-TI(1))/10.000 DO 399 J=2.10	

```
311 K1=K1+1
                                                                               305
    IF(K1.GE.12) GO TO 390
                                                                               306
    TOIF=T-CDF(I .K1)
                                                                               307
    IF(TCIF.GE.CDF(J1.12)) GG TO 350
                                                                               308
312 IF (TDIF.GT.CDF(J1.LI)) GO TO 340
                                                                               300
    LI=LI-1
                                                                               310
    IF(1-L1)312.312.390
                                                                               311
340 IF (LI .NE. 1) GO TO 360
                                                                               312
    T1=CDF(J1.1)
                                                                               313
    T2=CDF(J1.2)
                                                                               314
    T3=COF(J1.3)
                                                                               315
    Y1=0.00
                                                                               316
    Y2=0.0500
                                                                               317
    ¥3=0.1500
                                                                               318
    GC TC 361
                                                                               319
360 T1=CDF(J1.L1)
                                                                               320
    T2=CDF(J1.L1+1)
                                                                               321
    T3=CDF(J1.L1-1)
                                                                               322
    YI=A(LI)
                                                                               323
    Y2=A(LI+1)
                                                                               324
    Y3=A(LI-1)
                                                                               325
361 FOT=FOT+TERPOL(Y1.Y2.Y3.TDIF.T1.T2.T3)
                                                                               326
    GO TO 311
                                                                               327
350 F01=F01+1.D0
                                                                               328
    GC TC 311
                                                                               329
390 DENS(J)=0-100*FOT
                                                                               330
    IF(DENS(J).GE.1.DO) GO TO 380
                                                                               331
    IF(DENS(J).GT.O.DO) GO TO 399
                                                                               332
    DENS( J) = 0.00
                                                                               333
355 CENTINUE
                                                                               334
    GO TO 382
                                                                               335
380 CC 361 JJ=J.10
                                                                               336
    DENS(JJ)=1.00
                                                                               337
381 CONTINUE
                                                                               338
382 S(J1.2)=S(1.2)
                                                                               339
369 CONTINUE
                                                                               340
    CALL INVSTR(JI.DELTAT)
                                                                               341
    GO TO 379
                                                                               342
370 DO 372 JJ=1.12
                                                                               343
372 COF(J1.JJ)= COF(1.JJ) + COF(J1.12)
                                                                               344
    S(J1.2)=S(1.2)
                                                                               345
    GC TO 379
                                                                               346
376 DG 378 JJ=1.12
                                                                               347
378 CDF(J1.JJ)=CDF(J1.JJ) + CDF(1.12)
                                                                               348
    S(J1.2)= S(1.2)
                                                                               344
    RETURN
                                                                               350
                                                                               351
    END
    SUBROUTINE CCIRED(1.J1.J2.J3.J4.J12.J13)
                                                                               352
    IMPLICIT REAL+B(A-H.O-R.T-Z)
                                                                               353
    COMMON /BLKA/S.CDF.A.DENS.TI
                                                                               354
    INTEGER S
                                                                               355
    DIMENSION S(1000.4).CDF(1000.12).A(12).DENS(400).TI(2)
                                                                               356
    DO 406 J=1.2
                                                                               357
    L=1+(J-1)+11
                                                                               358
    T1=CDF(J3+L)+CDF(J4+L)+CDF(J12+L)
                                                                               359
    T2=COF(1.L)+COF(J2.L)+COF(J12.L)+COF(J4.L)
                                                                               360
    T3=CDF(1.L)+CDF(J1.L)+CDF(J12.L)
                                                                               361
    T4=CDF(1.L)+CDF(J13.L)
                                                                               362
    IF(T1.GE.T2.AND.T1.GE.T3.AND.T1.GE.T4) GO TO 403
                                                                               363
    IF(T2.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4) GC TO 404
                                                                               364
    IF(TJ.GE.TI.AND.TJ.GE.T2.AND.TJ.GE.T4) GO TO 405
                                                                               365
```

	TI(J)=T4	
	GC TO 406	
403	T1(J)=T1	
	GO TO 406	
404	T1(J)=12	
405	GO TO 406	
	T1(J)=13	
•06	CCNTINUE DELTAT=(T1(2)-TI(1))/10.0D0	
	DENS(1)=0.D0	
	DENS(11)=1.000	
	DO 470 J=2.10	
	I=II(1) + DFLUAT(J-1)+DELTAT	
	KI=1	
	L13=11	
	FCT=0.0	
10	K1=K1+1	
• •	IF(K1.GE.12) GQ TQ 460	
	TD1F13=T-CDF(1.K1)	
	IF(TDIF13.GE.CDF(J13.12)) GO TO 412	
11	IF(TDIF13.GT.CDF(J13.L13)) GO TO 413	
1	L13=L13-1	
	IF(1-L13)411.411.460	
12	F13=1.00	
	GO TO 415	
13	IF (L13.NE. 1) GO TO 491	
	T1=CDF(J13.1)	
	T2=CDF(J13,2)	
	T3=CDF(J13.3)	
	Y1=0.00	
	Y2=0.05D0	
	Y3=0.1500	
	GC TO 414	
91	T1=COF(J13,L13)	
	T2=CDF(J13,L13+1)	
	T3=CDF(J13,L13-1)	
	Y1=A(L13)	
	Y2=A(L13+1)	
	Y3=A(L13-1) F13=TERPOL(Y1.Y2.Y3.TDIF13.T1.T2.T3)	
••	IF (F13.LT.0.D0)F13=0.D0	
	IF(F13.GI-1.D0)F13=1.D0	
15	K12=1	
• •	L1=11	
	FGT2=0.00	
16	K12=K12+1	
-	1F(K12.LT.12) GO TO 418	
17	FCT=FCT+FGT2+F13	
	GO TO 410	
18	TD1F1=TD1F13-CDF(J12,K12)	
	IF(TDIF1.GE.CDF(J1.12)) GO TO 420	
19	IF(TDIF1.GT.CDF(J1.L1)) GO TO 421	
	L1=L1-1	
	IF(1-L1)419.419.417	
20	F1=1.00	
	GO TO 422	
21	IF (L1 .NE. 1) GO TO 482	
	T1=CDF(J1.1)	
	T2=COF(J1.2)	
	T3=CDF(J1.3)	
	Y1=0.D0	

	Y2=0.05D0	
	Y3=0.1500	
482	GC TG 427 T1=CDF(J1.L1)	
102	T2=CDF(J1.L1+1)	
	T3=COF(J1.L1-1)	
	Y1=A(L1)	
	Y2=A(L1+1)	
	Y3=A(L1-1)	
27	F1=TERPOL(Y1. Y2. Y3. TDIF1. T1. T2.	T3)
	IF(F1.LT.0.D0) F1=0.D0	
	IF(F1.GT.1.DO) F1=1.DO	
22	K4=1	
	L3=11	
	L2=11	
	FOT1=0.00	
23	K4=K4+1	
	IF(K4.LT.12) GO TO 425	
24	F012=F012+F1*F011	
	GC TO 416	
25	TD1F3=T-CDF(J4.K4)-CUF(J12.K12)	
	TD1F2=TD1F3-CDF(1.K1)	
	IF(TD1F2.GE.CDF(J2.12)) GO TO 430	
	IF(TDIF2.GT.CDF(J2.L2)) GO TO 431	
70	IF (1-L2) 426,426,424	
	F2=1.D0 GC TO 435	
	IF (L2 .NE. 1) GO TO 433	
	T1=CDF(J2.1)	
	12=CDF (J2.2)	
	T3=CDF(J2.3)	
	Y1=0.D0	
	Y2=0.05D0	
	Y3=0.15D0	
	GO TO 434	
133	11=CDF(J2.L2)	
	T2=CDF(J2.L2+1)	
	T3=CDF(J2.L2-1)	
	Y1=A(L2)	
	Y2=A(L2+1)	
	Y3=A(L2-1)	
134	F2=TERPOL(Y1. Y2. Y3. TDIF2. T1. T2.	(13)
	IF(F2 .LT.0.D0) F2=0.D0	
	IF(F2 .GT.1.D0) F2=1.D0	
135	IF(TDIF3.GE.COF(J3.12)) GO TO 440	
	IF(TDIF3.GT.CDF(J3.L3)) GO TO 445	
	L3=L3-1	
	IF(1-L3)436,436,424	
46	F3=1.00 GD TD 450	
	IF (L3 .NE. 1) GO TO 447	
45	T1=CDF(J3.1)	
	T2=CDF(J3.1)	
	T3=CDF(J3.2)	
	Y1=0.00	
	¥2=0.05D0	
	Y3=0.1500	SELECTION OF THE SELECT
	GO TO 449	
47	11=CDF(J3.L3)	
-	T2=CDF(J3.L3+1)	

```
T3=CDF(J3.L3-1)
                                                                              488
                                                                              489
    Y1=A(13)
    Y2=A(L3+1)
                                                                              490
                                                                              491
    Y3=A(L3-1)
449 F3=TERPOL(Y1, Y2, Y3, TOIF3, T1, T2, T3)
                                                                              492
    IF(F3 .LI.0.DJ) F3=0.D0
                                                                              493
    IF(F3.GT.1.D0) F3=1.D0
                                                                              494
450 FGT1=FCT1+F2*F3
                                                                              495
    GC TC 423
                                                                              4 96
460 DENS(J)=FCT+0.10-2
                                                                              497
    IF(DENS(J)-LT.0.DO) DENS(J)=0.D0
                                                                              498
    IF (DENS(J).GE.1.DO) GO TO 480
                                                                              499
470 CONTINUE
                                                                              500
480 DC 481 JJ=J.10
                                                                              501
    DENS(JJ)=1.DO
                                                                              502
ARL CONTINUE
                                                                              503
    S(J1.3)=S(J13.3)
                                                                              504
    5(11.2)=5(1.2)
                                                                              505
    CALL INVSTR(JI.DELTAT)
                                                                              506
    RETURN
                                                                              507
                                                                              508
    ENC
    SUBROUTINE CC2RED(1.J1.J2.J3.J4.J8.J12)
                                                                              509
    IMPLICIT REAL+8(A-H.G-R.T-Z)
                                                                              510
    CCMMON /BLKA/S.CDF.A.DENS.TI
                                                                              511
    INTEGER S
                                                                              512
    DIMENSION S(1000.4).CDF(1000.12).A(12).DENS(400).TI(2)
                                                                              513
    00 510 J=1.2
                                                                              514
                                                                              515
    L=1+(J-1)011
    TI=COF(I.L)+COF(JI.L)+COF(JI2.L)
                                                                              516
    T2=COF(1.L)+COF(J2.L)+COF(J4.L)+COF(J12.L)
                                                                              517
    T3=CDF(1.L)+CDF(J2.L)+CDF(J8.L)
                                                                              518
    T4=CDF(J3.L)+CDF(J4.L)+CDF(J12.L)
                                                                              519
    T5=COF(J3.L)+CDF(J8.L)
                                                                              520
    IF(T1.GE.T2.AND.T1.GE.T3.AND.T1.GE.T4.AND.T1.GE.T5) GO TO 501
                                                                              521
    IF(T2.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4.AND.T2.GE.T5) GO TO 502
                                                                              522
    IF(T3.GE.T1.AND.T3.GE.T2.AND.T3.GE.T4.AND.T3.GE.T5) GO TO 503
                                                                              523
    IF(T4.GE.T1.AND.T4.GE.T2.AND.T4.GE.T3.AND.T4.GE.T5) GO TO 504
                                                                              524
    T1(J)=15
                                                                              525
    GO TO 510
                                                                              526
501 TI(J)=T1
                                                                              527
    GC TO 510
                                                                              528
502 TI(J)=T2
                                                                              529
    GO TO 510
                                                                              530
503 T1(J)=T3
                                                                              531
    GO TO 510
                                                                              532
504 TI(J)=14
                                                                              533
510 CCATINUE
                                                                              534
    DELTAT=(T1(2)-T1(1))/10.000
                                                                              535
    DENS(1)=0.00
                                                                              536
    DENS(11)=1.000
                                                                              537
    DC 570 J=2.10
                                                                              53A
    T=T((1) + DFLOAT(J-1)+DELTAT
                                                                              539
    K1=1
                                                                              540
    FOT=0.00
                                                                              541
511 KI=KI+1
                                                                              542
    IF (KI.GE.12) GO TO 560
                                                                              543
                                                                              544
    K12=1
                                                                              545
    L1=11
    FOT3=0.00
                                                                              546
512 K12=K12+1
                                                                              547
                                                                              548
    IF(K12.LT.12) GO TO 514
```

513 FCT=FCT+FCT3	549
GQ TQ 511	550
514 TCIF1=T-CGF(1.KI)-CDF(J12.K12)	551
IF(TDIF1.GE.CDF(J1.12)) GG TO 517	552
515 IF(TDIF1.GT.CDF(J1.L1)) GO TO 518	553
Li=Li-i	554
IF(1-L1)515,515,513	555
517 F1=1.00	556
GO TO 519	557
518 IF (L1 .NE. 1) GO TO 971	558
T1=CDF(J1.1)	559
T2=CDF(J1.2) T3=CDF(J1.3)	560 561
Y1=0.00	562
Y2=0.0500	563
Y3=0.15D0	564
GU TU 972	565
971 T1=CDF(J1+L1)	566
T2=CDF(J1.L1+1)	567
T3=CDF(J1.L1-1)	568
Y1=A(L1)	569
Y2=A(L1+1)	570
Y3=A(L1-1)	571
972 F1=TERPOLIYI, Y2. Y3. TDIF1. T1. T2.	
IF(F1.LT.0.D0) F1=0.D0	573
IF(F1.GT.1.00) F1=1.00	574
519 K3=1	575
FC12=0.00	576
520 K3=K3+1	577
IF(K3.LT.12) GO TO 521	578
F0T3=FCT3+F1*F0T2	579
GO TO 512	580
521 K2=1	561
FCT1=0.00	582
L4=11	563
L8=11	584
522 K2=K2+1	585
IF(K2.LT.12) GO TO 526	586
525 FCT2=FCT2+FOT1	587
GO TO 520	588
526 T1=T-CDF(J3,K3)	589
T2=T-CDF(1.K1)-CDF(J2.K2)	590
IF(T2-T1)528.527.527	591
528 TD1F8=T2	592
GO TO 530	593
527 TC1F8=T1	594
TDIF4=TDIF8-CDF(J12.K12)	595
530 IF(TCIF4.GE.CDF(J4.12) GO TO 535	596
	597
L4=L4-1	598
535 F4=1.CO	599
GC TO 541	600
540 IF (L4 .NE. 1) GO TO 973	601
T1=CDF(J4+1)	602
T2=CDF(J4.2)	603
T3=CDF(J4.3)	604
Y1=0.D0	605
Y2=0.0500	606
Y3=0.1500 GO TO 974	607
973 T1=CDF(J4-L4)	609
7/3 /1-00/(34)04)	609

```
610
    T2=CDF(J4.L4+1)
    T3=CDF(J4-L4-1)
                                                                                611
                                                                                612
    Y1=A(L4)
                                                                                613
    Y2=A(L4+1)
    Y3=A(L4-1)
                                                                                614
974 F4=TERPOL(Y1. Y2. Y3. TDIF4. T1. T2. T3)
                                                                                615
    IF(F4 .LT.0.DO) F4=0.00
                                                                                616
    IF (F4 .GT.1.DO) F4=1.DO
                                                                                617
541 IF(TDIF8.GE.CDF(J8.12)) GO TO 545
                                                                                618
542 IF(TDIF8.GT.CDF(J8.L8)) GO TC 550
                                                                                619
                                                                               620
    L8=L8-1
    IF(1-L8) 542.542.525
                                                                                621
545 F8=1.D0
                                                                               622
    GO TO 551
                                                                                623
550 IF (LB .NE. 1) GO TO 975
                                                                                624
    T1=CDF(J8.1)
                                                                                625
    12=CDF(J8.2)
                                                                                626
    T3=CDF(J8.3)
                                                                                627
    Y1=0.C0
                                                                                628
    Y2=0.0500
                                                                                629
    ¥3=0.1500
                                                                                630
    GO TO 576
                                                                                631
975 T1=CDF(J8.L3)
                                                                                632
    T2=CDF(J8.L8+1)
                                                                                633
    73=CDF(J8.L8-1)
                                                                                634
    Y1=A(L8)
                                                                                635
    Y2=A(L8+1)
                                                                                636
    Y3=A(L8-1)
                                                                                637
976 F8=TERPOL(Y1. Y2. Y3. TOLF8. T1. T2. T3)
                                                                                638
    IF(F8 .LT.0.00) F8=0.00
                                                                                639
    IF(F8. GT.1.DO) F8=1.00
                                                                                640
551 FOT1=FOT1+F4*F8
                                                                                641
    GG TO 522
                                                                                642
560 DENS(J)=FCT+0-10-3
                                                                                643
    IF(DENS(J).LT.O.DO) DENS(J)=0.DO
                                                                                644
    IF(DENS(J).GE.1.DO) GO TO 580
                                                                                645
570 CONTINUE
                                                                                646
580 DO 581 JJ=J.10
                                                                                647
    DEHS(JJ)=1.00
581 CCNTINUE
                                                                                649
    S(J1.2)=S(1.2)
                                                                                650
    S(J1.3)=S(J12.3)
                                                                                651
    CALL INVSTR(JI . DELTAT)
                                                                                652
    RETURN
                                                                                653
    END
                                                                                654
    SUBROLTINE CC3RED(1. J1. J2. J3. J5. J6. J7)
                                                                                655
    IMPLICIT REAL+8(A-H.O-R.T-Z)
                                                                                656
    COMMON /BLKA/S.COF.A.DENS.TI
                                                                                657
    INTEGER S
                                                                                658
    DIMENSION S(1000.4).CDF(1000.12).A(12).DENS(400).T1(2)
                                                                                659
    DO 610 J=1.2
                                                                                660
    L=1+(J-1)+11
                                                                                661
    T1=CDF(1,L)+CDF(J1.L)+CDF(J6.L)
                                                                                662
    T2=CDF(1.L)+CDF(J2.L)+CDF(J7.L)
                                                                                663
    T3=CDF(1,L)+CDF(J1,L)+CDF(J5,L)+CDF(J7,L)
                                                                                664
    T4=CDF(J3.L)+CDF(J7.L)
                                                                                665
    IF(TI.GE.T2.AND.TI.GE.T3.AND.TI.GE.T4) GO TO 602
                                                                                666
    IF(T2.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4) GG TO 603
                                                                                667
    IF(T3.GE.T1.AND.T3.GE.T2.AND.T3.GE.T4) GO TO 604
                                                                               668
    T1(J)=T4
                                                                                669
    GG TO 610
                                                                                670
```

.

602	TI(J)=TI	671
407	GO TO 610	672
603	T1(J)=T2 G0 T0 610	673
604	T1(J)=T3	674 675
004	GO TO 610	
610	CENTINUE	676
010	DELTAT=(TI(2)-TI(1))/10.0D0	677 678
	DENS(1)=0.DO	679
	DENS(11)=1.0D0	680
	00 670 J=2,10	681
	T=TI(1) + DFLOAT(J-1) +DELTAT	682
	K7=1	683
	L3=11	684
	FCT=0.D0	685
615	K7=K7+1	686
	IF(K7.GE.12) GO TO 660	687
	TD1F3=T-CDF(J7,K7)	688
	IF(TD1F3.GE.CDF(J3.12)) GO TO 625	689
620	IF(TDIF3.GT.CDF(J3.L3)) GO TC 622	690
	L3=L3-1	691
	IF(1-L3)620,620,660	692
622	IF (L3 .NE. 1) GO TO 671	693
	T1=CDF(J3.1)	694
	T2=CDF(J3.2)	695
	T3=CDF(J3.3)	696
	Y1=0.C0	697
	Y2=0.05D0	698
	Y3=0.15D0	699
	GO TO 672	700
671	T1=CDF(J3,L3)	701
	T2=CDF(J3+L3+L)	702
	T3=CDF(J3,L3-1)	703
	Y1=A(L3)	704
	Y2=A(L3+1)	705
	Y3=A(L3-1)	706
672	F3=TERPOL(Y1, Y2, Y3, TDIF3, T1, T2, T3)	707
	IF(F3 .LT.0.D0) F3=0.D0	708
	IF(F3.GT.1.00) F3=1.D0	709
	GO TO 626	710
The state of the s	F3=1.00	711
626	KI=1	712
	L2=11	713
427	FOT2=0.00 KI=KI+1	714
621	1F(K1.LT.12) GO TO 630	715
628	FOT=FOT+F3*FOT2	716
020	GC TO 615	717 718
630	TDIF2=TDIF3-CDF(1.KI)	719
	IF(TDIF2.GE.COF(J2.12)) GO TO 635	720
631	IF(TDIF2.GT.CDF(J2.L2)) GO TO 636	721
	L2=L2-1	722
	IF(1-L2)631,631,628	723
635	F2=1.00	724
	GO TC 637	725
636	IF (L2 .NE. 1) GO TO 673	726
	T1=CDF(J2.1)	727
	T2=CCF(J2,2)	728
	T3=CDF(J2.3)	729
	Y1=0.00	730
	Y2=0.0500	731

¥3=0.		732
GO TO		733
673 T1=C0		734
	F(J2.L2+1)	735 736
Y1=A(F(J2.L2-1)	737
	L2+1)	738
	L2-1)	739
	RPOL (Y1. Y2. Y3. TDIF2. T1. T2. T3)	740
IF (F2	.LT.0.D0) F2=0.D0	741
IF(F2	.GT.1.DO) F2=1.DO	742
637 K1=1		743
FCT1=		744
L5=11		745
L6=11		746
	•LT•12) GU TO 645	76
	FCT2+F2+F0T1	70.0
GO TO		750
	=TDIF2-CDF(J1.K1)	751
TD1F6	=T-CDF(1,K1)-CDF(J1,K1)	752
	1F5.GE.CDF(J5.12)) GO TO 650	753
	IF5.GT.CDF(J5.L5)) GO TO 647	754
L5=L5		755
	L51646,646,641	756 757
	5 •NE• 1) GO TO 675	758
	F(J5•2)	759
	F(J5.3)	760
Y1=0.		761
Y2=0.	0500	762
¥3=0.	1500	763
GO TO		764
675 T1=CD		765
	F(J5,L5+1)	766
Y1=A(F(J5+L5-1)	767 768
	L5+1)	769
	L5-1)	770
	RPOL(Y1, Y2, Y3, TOIF5, T1, T2, T3)	771
IF (F5	.LT.0.D0) F5=0.D0	772
IF (F5	.GT.1.DO) F5=1.DO	773
GO TO		774
650 F5=1.		775
	1F6.GE.CDF(J6.12)) GO TO 655	776
652 IF(10	1F6.GT.CDF(J6.L6)) GO TO 656	777 778
	L6)652.652.641	779
655 F6=1.		780
GO TO		781
	6 .NE. 1) GO TO 677	782
TI=CD	F(J6.1)	783
	F(J6.2)	784
	F(J6.3)	785
Y1=0.		786
Y2=0. Y3=0.		787 788
GO TO		789
677 T1=C0		790
The state of the s	F(J6.L6+1)	791
The state of the s	F(J6.L6-1)	792

	Y1=A(L6)	793
	Y2=A(L6+1) Y3=A(L6-1)	794 795
678	F6=TERPCL(Y1. Y2. Y3. TDIF6. T1. T2. T3)	796
0.0	IF (F6.LT.0.D0)F6=0.D0	797
	IF(F6.GT.1.DO) F6=1.DO	798
659	FOT1=FOT1+F5*F0	799
	GO TO 640	800
660	DENS(J)= FOT+0.10-2	801
	1F(DENS(J).LT.0.DO) DENS(J)=0.DO	802
	IF(DENS(J).GE.1.DO) GO TO 680	803
	CONTINUE	804
680	DENS(JJ)=1.00	805 806
681	CONTINUE	807
	S(J1.2)=S(1.2)	808
	S(J1.3)=S(J6.3)	809
	CALL INVSTR(JI.DELTAT)	810
	RETURN	811
	END	812
	SUBROUTINE CHIREO(1.J1.J2.J3.J4.J9.J10.J11)	813
	IMPLICIT REAL+8(A-H.O-R.T-Z)	814
	CCMMON /BLKA/S, COF, A. DENS, TI	815
	INTEGER S	816
	DIMENSION S(1000.4).CDF(1000.12).A(12).DENS(400).TI(2)	817
	CO 710 J=1.2 L=1+(J-1)+11	818
	T1=CDF([,L)+CDF(J1,L)	820
	T2=CDF([,L)+CDF(J2,L)+CDF(J4,L)	821
	T3=CDF(J10+L)+CDF(J11+L)	822
	T4=CDF(J9,L)+CDF(J10,L)+CDF(J4,L)	823
	T5=CDF(J3.L)+CDF(J4.L)	824
	IF(T1.gE.T2.AND.T1.gE.T3.AND.T1.GE.T4.AND.T1.GE.T5) GO TO 701	825
	IF(T2.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4.ANC.T2.GE.T5)GD TO 702	826
	IF(T3.GE.T1.AND.T3.GE.T2.AND.T3.GE.T4.AND.T3.GE.T5) GC TO 703	827
	IF(T4.GE.T1.AND.T4.GE.T2.AND.T4.GE.T3.AND.T4.GE.T5) GO TO 704	828
	T1(J)=T5	829
	GC TO 710	830
701	T1(J)=T1 GG TO 710	831 832
702	T1(J)=T2	833
	GC TC 710	834
703	T1(J)=T3	835
	GO TO 710	836
704	TI(J)=T4	837
710	CONTINUE	838
	DEAS(1)=0.DO	839
	DENS(11)=1.000	840
	DELTAT=(T1(2)-T1(1))/10.0D0	841
	DO 790 J=2.10	842
	T=TI(1) + DFLOAT(J-1) DELTAT	843
		845
		846
715	K4=K4+1	847
	IF(K4.GE.12) GO TO 785	848
	TDIF3=T-CDF(J4,K4)	849
	IF(TCIF3.GE.CDF(J3.12)) GO TO 720	850
716	IF(TDIF3.GT.CDF(J3.L3)) GO TG 717	851
	L3=L3-1	852
	IF(1-L3)716,716,785	853

717	IF (L3 .NE. 1) GO TO 791		854
	T1=COF(J3.1)		855
	T2=CDF(J3.2)		856
	T3=C0F(J3,3)		857
	Y1=0.D0		858
	Y2=0.0500		859
	Y3=0.1500 GD TO 792		861
701	T1=CDF(J3.L3)		862
,,,,	T2=CDF(J3.L3+1)		863
	T3=CDF(J3.L3-1)		864
	Y1=A(L3)		865
	Y2=A(L3+1)		866
	Y3=A(L3-1)		867
792	F3=TERPOL(Y1. Y2. Y3, TDIF3. T1, T2. T3)		868
	IF(F3 -LT.0.D0) F3=0.D0		869
	IF(F3.GT.1.D0) F3=1.D0		870
	GG TO 721		871
720	F3=1.00		872
721	K1=1		873
	FOT1=0.DO		874
	L2=11		875
	L1=11		876
725	K1=K1+1		877
	IF(KI.LT.12) GO TO 730		878
726	FGI=FCI+FOT1+F3		879
	GO TO 715		880
730	TD1F2=TD1F3-CDF(1.K1)		881
	TDIF1=T-CDF(1,K1)		882
	IF(TDIF2.GE.CDF(J2.12)) GO TO 732		883
/31	IF(1DIF2.GT.CDF(J2.L2)) GO TO 733		884
	L2=L2-1 IF(1-L2)731.731.726		885
712	F2=1.00		887
,,,	GO TO 735		888
733	IF (L2 .NE. 1) GO TO 793		889
	T1=CDF(J2+1)		890
	T2=CDF(J2+2)		891
	T3=CDF(J2.3)		892
	Y1=0.D0		893
	Y2=0.05D0		894
	¥3=0.15D0		895
	GC TO 974		896
793	T1=CDF(J2.L2)		897
	T2=CDF(J2+L2+1)		898
	T3=CDF(J2,L2-1)		899
	Y1=A(L2)		900
	Y2=A(L2+1)	1	
	YJ=A(L2-1)		902
974	F2=TERPOL(Y1, Y2, Y3, TDIF2, T1, T2, T3)		903
	IF(F2 .LT.0.00) F2=0.D0		904
	IF(F2 .GT.1.00) F2=1.D0		905
	IF(TDIF1.GE.CDF(J1.12)) GO TO 737		906
736	IF(TD1F1.GT.CDF(J1.L1)) GO TO 738		907
	L1=L1-1 IF(1-L1)736,736,726		908
777	f1=1.00		910
, ,,	GC TO 740		911
770	IF (L1 •NE• 1) GO TO 795		912
, 30	T1=CDF(J1+1)		913
	T2=CDF(J1.2)		914
	12-CDF (31 12 /		-14

	T3=CDF(J1.3)	
	V1=0.00 V2=0.05D0	
	Y3=0.1500	
	GO TO 796	
95	T1=CDF(J1.L1)	
••	T2=CDF(J1.L1+1)	lel a
	T3=CCF(J1.L1-1)	
	Y1=A(L1)	
	Y2=A(L1+1)	
	Y3=A(L1-1)	
96	F1=TERPOL(Y1. Y2. Y3. TOIF1. T1. T2. T3)	
-	IF(F1-LT-0-00) F1=0-00	
	IF(F1.GT.1.D0) F1=1.D0	
40	K10=1	
	FC12=0.00	
	L9=11	
	L11=11	
41	K10=K10+1	
	IF(K10.LT.12) GO TO 743	
42	FCT1=FOT1+F2+F1+FOT2	
	GO TO 725	
43	TCIF9=TCIF3-CDF(J10,K10)	
	TDIF11=T-CDF(J10.K10)	
	LF(TD1F9.GE.COF(J9.121) GO TO 747	
44	IF(TDIF9.GT.COF(J9.L9)) GO TO 748	
	L9=L9-1	
	IF(1-L9)744,744,742	
47	F9=1.00	
	GC TC 750	
48	IF (L9 .NE. 1) GO TO 797	
	T1=CCF(J9.1)	
	T2=CDF(J9.2)	
	13=CDF(J9.3)	
	Y1=0.00	
	¥2=0.05D0	
	Y3=0.1500	
	GC 10 798	
97	T1=CDF(J9.L9)	
	72=CDF(J9.L9+1)	
	T3=CDF(J9.L9-1)	
	Y1=A(L9)	
	Y2=A(L9+1)	
	Y3=A(L9-1)	
98	F9=TERPOL(Y1, Y2, Y3, TDIF9, T1, T2, T3)	
	IF(F9.LT.0.D0) F9=0.D0	
	IF(F9.GT.1.DO) F9=1.DO	104
	IF(TDIF11.GE.COF(J11.12)) GO TO 755	
21	IF(TDIFIL.GT.CDF(JII.LII)) GG TO 756	
	L11=L11-1	
	IF(1-L11)751.751.742	600
25	F11=1.00 GO TO 760	
90	IF (L11.NE. 1) GO TO 799 T1=CDF(J11.1)	
	T2=CDF(J11,2) T3=CDF(J11,3)	618
	Y1=0.00	
	Y2=0.05C0	
	12-010300	
	¥3=0.1500	

```
799 T1=CDF(J11.L11)
                                                                              976
    12=CDF(J11.L11+1)
                                                                              977
    T3=CDF(J11.L11-1)
                                                                              978
    Y1=A(L11)
                                                                              979
    Y2=A(L11+1)
                                                                              980
                                                                              981
    Y3=A(L11-1)
800 FII=TERPOL(YI, Y2, Y3, TDIFII, T1, T2, T3)
                                                                              982
    IF(F11.LT.0.DO) F11=0.DO
                                                                              FHP
    IF (F11.GT.1.DO) F11=1.DO
                                                                              984
760 FCT2=F0T2+F9+F11
                                                                              985
                                                                              986
    GO TG 741
785 DENS(J) = FOT+0.10-2
                                                                              987
    IF (DENS(J).LT.0.DO) DENS(J)=0.DO
                                                                              988
    IF(DENS(J).GE.1.DO) GO TO 780
                                                                              989
790 CONTINUE
                                                                              990
780 DG 781 JJ=J.10
                                                                              991
    DENS(JJ)=1.00
                                                                              992
                                                                              993
781 CENTINUE
    S(J1.2)=S(1.2)
                                                                              994
    CALL INVSTR(J1.DELTAT)
                                                                              995
    RETURN
                                                                              996
    END
                                                                              997
    SUBROUTINE D#2RED(1, J1, J2, J3, J4, J8, J10, J11)
                                                                              998
    IMPLICIT REAL+8(A-H.O-R.T-Z)
                                                                              999
    COMMON /BLKA/S.CDF.A.DENS.TI
                                                                             1000
                                                                             1001
    DIMENSION S(1000.4).CDF(1000.12).A(12).DENS(400).TI(2)
                                                                             1002
    DG 810 J=1.2
                                                                             1003
    L=1+(J-1)+11
                                                                             1004
    TI=CDF(I.L)+CDF(J1.L)
                                                                             1005
    12=CDF(J2.L)+CDF(I.L)+CDF(J4.L)
                                                                             1006
    T3=CDF(1.L)+CDF(J2.L)+CDF(J8.L)+CDF(J11.L)
                                                                             1007
    T4=CDF(J3.L)+CDF(J4.L)
                                                                             1008
    T5=CDF(J3.L)+CDF(J8.L)+CDF(J11.L)
                                                                             1009
    T6=CDF(J10.L)+CDF(J11.L)
                                                                             1010
    IF(T1.GE.T2.AND.T1.GE.T3.AND.T1.GE.T4.AND.T1.GE.T5.AND.T1.GE.T6)
                                                                             1011
   IGC TO BOL
                                                                             1012
    IF(12.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4.AND.T2.GE.T5.AND.T2.GE.T6)
                                                                             1013
   1GO TO 802
                                                                             1014
    1F(T3.GE.T1.AND.T3.GE.T2.AND.T3.GE.T4.AND.T3.GE.T5.AND.T3.GE.T6)
                                                                             1015
   1GC TO 803
                                                                             1016
    IF (T4.GE.T1.AND.T4.GE.T2.AND.T4.GE.T3.AND.T4.GE.T5.AND.T4.GE.T6)
                                                                             1017
   1GO TO 804
                                                                             1018
    IF(T5.GE.T1.AND.T5.GE.T2.AND.T5.GE.T3.AND.T5.GE.T4.AND.T5.GE.T6)
                                                                             1019
   1GC TC 805
                                                                             1020
    T1(J)=T6
                                                                             1021
    GC TO 810
                                                                             1022
801 TI(J)=T1
                                                                             1023
    GO TO 810
                                                                             1024
802 TI(J)=T2
                                                                             1025
    GO TO 810
                                                                             1026
803 T1(J)=T3
                                                                             1027
    GO TO 810
                                                                             1028
804 TI(J)=T4
                                                                             1029
    GO TO 810
                                                                             1030
805 TI(J)=15
                                                                             1031
810 CONTINUE
                                                                             1032
    DENS(1)=0.00
                                                                             1033
    DENS(11)=1.000
                                                                             1034
    DELTAT=(T1(2)-T1(1))/10.000
                                                                             1035
    DO 890 J=2.10
                                                                             1036
```

	T=TI(1) + DFLUAT(J-1) *DELTAT K14=1	1037
	L10=11	1039
	FOT=0.D0	1040
815	K11=K11+1	1041
	IF(K11.GE.12) GO TO 885	1042
	TDIF10=T-COF(J11.K11)	1043
	IF(TDIF10.GE.CDF(J10.12)) GO TO 817	1044
816	IF(TDIF10.GT.CDF(J10.L10)) GO TO 818	1045
	L10=L10-1	1046
	IF(1-L10)810.816.885	1047
817	F10=1.00	1048
	GO TC 820 IF (L10.NE. 1) GC TO 791	1049
010	T1=CDF(J10.1)	1050
	T2=CDF (J10,2)	1052
	T3=CDF(J10,3)	1053
	Y1=0.00	1054
	Y2=0.05D0	1055
	¥3=0.15D0	1056
	GO TO 792	1057
791	T1=CCF(J10.L10)	1058
	T2=CDF(J1G.L10+1)	1059
	T3=CCF(J10.L10-1)	1060
	Y1=A(L10)	1061
	Y2=A(L10+1)	1062
	Y3=A(L10-1)	1063
792	F10=TERPOL(Y1. Y2. Y3. TDIF10. T1. T2. T3)	1064
	IF(F10.LT.0.D0)F10=0.D0	1065
820	IF(F10.GT.1.D0) F10=1.D0 KI=1	1066
820	FOT3=0.00	1068
	L1=11	1069
821	K1=K1+1	1070
	IF(KI-LT-12) GO TO 823	1071
822	FCT=FCT+F10+FOT3	1072
	GC TO 815	1073
823	TDIF1=T-CDF(1.K1)	1074
	IF(TDIF1.GE.CDF(J1.12)) GO TO 825	1075
824	IF(TDIF1.GT.CDF(J1.L1)) GO TO 826	1076
	Li=Li-1	1077
	IF(1-L1)824.824.822	1078
825	F1=1.C0	1079
	60 10 830	1080
826	IF (L1 •NE• 1) GO TO 893 T1=CDF(J1•1)	1081
	T2=CDF(J1.2)	1083
	T3=CDF(J1.3)	1084
	Y1=0.00	1085
	Y2=0.05D0	1086
	¥3=0.15D0	1087
	GO TO 894	1089
893	T1=CCF(J1,L1)	1089
	T2=CDF(J1,L1+1)	1090
	T3=COF(J1.L1-1)	1091
	Y1=A(L1)	1092
	Y2=A(L1+1)	1093
	Y3=A(L1-1)	1094
894	F1=TEKPOL(Y1, Y2, Y3, TOIF1, T1, T2, T3)	1095
	IF(F1.GT.1.DO) F1=0.DO IF(F1.GT.1.DO) F1=1.DO	1098
		.057

830	K8=1	1098
	F012=0.00	1099
831	KE=K8+1	1100
	IF(K8.LT.12) GO TO 832	1101
	F0T3=F0T3+F1+F0T2	1102
	GO TO 821	1103
832	K4=1	1104
	FCT1=0.00	1105
	L2=11 L3=11	1106
077	K4=K4+1	1108
633	IF(K4.GE.12) GU TO 840	1109
	T1=T-CDF(J4-K4)	1110
	T2=TD1F10-CDF(J8.K8)	1111
	IF(T1-T2)834,835,835	1112
874	TD1F3=T1	1113
034	GO TO 836	1114
P35	TD1F3=12	1115
	TDIF2=TDIF3-CDF(1,KI)	1116
	IF(TDIF2.GE.COF(J2.12)) GO TO 838	1117
837	1F(TD1F2.GT.CDF(J2.L2)) GO TC 839	1118
	L2=L2-1	1119
	IF(1-L2)837.837.840	1120
840	FOT2=FOT2+FOT1	1121
	GG TO 831	1122
858	F2=1.00	1123
	GO TO 845	1124
839	IF (L2 .NE. 1) GO TO 895	1125
	T1=CDF(J2.1)	1126
	T2=CDF(J2.2)	1127
	T3=CDF(J2.3)	1128
	Y1=0.00	1129
	Y2=0.05D0	1130
	Y3=0.1500	1131
	GO TO 896	1132
895	T1=CDF(J2.L2)	1133
	T2=C0F(J2,L2+1)	1134
	T3=COF(J2,L2-1)	1135
	Y1=A(L2)	1136
	Y2=A(L2+1) Y3=A(L2-1)	1137
806	F2=TERPOL(Y1, Y2, Y3, TDIF2, T1, T2, T3)	1139
0,0	1F(F2 .LI.0.D0) F2=0.D0	1140
	IF(F2 •GT •1 • D0) F2=1 • D0	1141
245	IF(TDIF3.GE.CDF(J3.12)) GO TO 850	1142
1100	IF(TDIF3.GT.CDF(J3.L3)) GO TO 851	1143
	L3=L3-1	1144
	IF(1-L3)846,846,840	1145
850	F3=1.00	1146
	GO TO 860	1147
851	IF (L3 .NE. 1) GO TO 897	1148
	T1=CDF(J3.1)	1149
	T2=CDF(J3.2)	1150
	T3=C0F(J3,3)	1151
	Y1=0.00	1152
	Y2=0.0500	1153
	¥3=0.15D0	1154
	GC 1C 898	1155
857	T1=CDF(J3+L3) T2=CDF(J3+L3+L)	1156
	77-07(17)	1157
	T3=COF(J3.L3-1)	1158

```
1159
      Y1=A(L3)
      Y2=A(L3+1)
                                                                              1160
      Y3=A(L3-1)
                                                                              1161
  858 F3=TERPOL(Y1, Y2, Y3, TDIF3, T1, T2, T3)
                                                                              1162
      IF(F3 .LT.0.D0) F3=0.D0
                                                                              1163
      IF(F3.GT.1.D0) F3=1.D0
                                                                              1164
  860 FOT1=FOT1+F3*F2
                                                                              1165
      GO TO 833
                                                                              1166
  885 DENS(J)= FOT+0.10-3
                                                                              1167
      IF (DENS(J).LT.0.DO) DENS(J)=0.DO
                                                                              1168
      IF(DENS(J).GE.1.DO) GO TO 880
                                                                              1169
  890 CONTINUE
                                                                              1170
  880 DO 881 JJ=J.10
                                                                              1171
      DENS(JJ)=1.00
                                                                              1172
  881 CONTINUE
                                                                              1173
      S(J1.2)=S(1.2)
                                                                              1174
      CALL INVSTR(JI.DELTAT)
                                                                              1175
      RETURN
                                                                              1176
                                                                              1177
      SUBROUTINE WORED (1.J1.J2.J3.J4)
                                                                              1178
      IMPLICIT REAL +8(A-H.O-R.T-Z)
                                                                              1179
      COMMON /BLKA/S.CDF.A.DENS.TI
                                                                              1180
      INTEGER S
                                                                              1181
      DIMENSION S(1000.4).CDF(1000.12).A(12).DENS(400).TI(2)
                                                                              1182
                 . . . . . . . .
C
           .
                                                                              1183
C
                                                                              1184
C
                                                                              1185
      00 910 J=1.2
                                                                              1186
      L=1+(J-1)+11
                                                                              1187
      T1=CDF(I.L)+CDF(J1.L)
                                                                              1188
      T2=CDF(1.L)+CDF(J2.L)+CDF(J4.L)
                                                                              1189
      T3=CDF(J3.L)+CDF(J4.L)
                                                                              1190
      IF(T1-T2) 901.901.902
                                                                              1191
  901 IF(T2-T3) 903,903,904
                                                                              1192
  902 IF(T1-T3)903.903.905
                                                                              1193
                                                                              1194
  903 TI(J)=13
      GO TO 910
                                                                              1195
  904 TI(J)=T2
                                                                               1195
      GO TO 910
                                                                              1197
  905 TI(J)=T1
                                                                              1198
  SIO CONTINUE
                                                                              1199
                                                                              1200
      DENS(1)=0.00
                                                                              1201
      DENS(11)=1.000
      DELTAT=(T1(2)-T1(1))/10.000
                                                                              1202
      DO 990 J=2.10
                                                                              1203
      T=TI(1)+DFLOAT(J-1)+DELTAT
                                                                              1204
                                                                              1205
      K4=1
                                                                              1206
      FCT=0.CO
                                                                              1207
      L3=11
                                                                              1208
  915 K4=K4+1
      IF(K4.GE.12) GO TO 985
                                                                              1209
      TDIF3=T-CDF(J4.K4)
                                                                              1210
      IF(TDIF3.GE.COF(J3.12)) GO TO 918
                                                                              1211
                                                                              1212
  916 IF(TDIF3.GT.COF(J3.L3)) GO TO 917
                                                                              1213
      L3=L3-1
      1F(1-L3)916.916.985
                                                                              1214
C
                                                                              1215
C
                                                                              1216
                                                                              1217
C
                                                                              1218
  917 IF (L3 .NE. 1) GO TO 951
                                                                              1219
```

```
T1=CDF(J3.1)
                                                                                 1220
      T2=CDF(J3.2)
                                                                                 1221
      13=CDF(J3,3)
                                                                                 1222
      Y1=0.00
                                                                                 1223
      ¥2=0.0500
                                                                                 1224
      ¥3=0.1500
                                                                                 1225
      GO TO 952
                                                                                 1226
  951 T1=CDF(J3.L3)
                                                                                 1227
      T2=CDF(J3.L3+1)
                                                                                 1228
      T3=CDF(J3.L3-1)
                                                                                 1229
      Y1=A(L3)
                                                                                 1230
                                                                                 1231
      Y2=A(L3+1)
      Y3=A(L3-1)
                                                                                 1232
  952 F3=TERPOL(Y1. Y2. Y3. TDIF3. T1. T2. T3)
                                                                                 1233
      IF(F3 .LT.0.D0) F3=0.D0
                                                                                 1234
      IF(F3.GT.1.DO) F3=1.DO
                                                                                 1235
            .
                   .
                                                                                 1236
C
C
                                                                                 1237
C
                                                                                 1238
                                                                                 1239
      GC TO 920
                                                                                 1240
  918 F3=1.00
                                                                                 1241
  920 KI=1
                                                                                 1242
      FOT1=0.00
                                                                                 1243
      L2=11
                                                                                 1244
                                                                                 1245
      L1=11
                                                                                 1246
  921 KI=KI+1
      IF(KI.LT.12) GO TO 925
                                                                                 1247
  922 FOT=FOT+FOT1+F3
                                                                                 1248
      GC TC 915
                                                                                 1249
  925 TD1F2=TD1F3-CDF(1.K1)
                                                                                 1250
      TDIF1=T-COF(1.KL)
                                                                                 1251
      IF(TDIF2.GE.COF(J2.12)) GOTO 928
                                                                                 1252
  926 IF(TDIF2.GT.CDF(J2.L2)) GG TO 927
                                                                                 1253
      L2=L2-1
                                                                                 1254
      IF(1-L2)926,926,922
                                                                                 1255
C
                                                                                 1256
C
                                                                                 1257
C
                                                                                 1258
                                                                                 1259
  927 IF (L2 .NE. 1) GO TO 961
                                                                                 1260
      T1=CCF(J2.1)
                                                                                 1261
      T2=CDF(J2.2)
                                                                                 1262
      T3=CDF(J2.3)
                                                                                 1263
      Y1=0.C0
                                                                                 1264
      Y2=0.0500
                                                                                 1265
      ¥3=0.15D0
                                                                                 1266
      GO TO 962
                                                                                 1267
  961 T1=CDF(J2.L2)
                                                                                 1268
      T2=CDF(J2.L2+1)
                                                                                 1269
      T3=CDF(J2.L2-1)
                                                                                 1270
      Y1=A(L2)
                                                                                 1271
      Y2=A(L2+1)
                                                                                 1272
      Y3=A(L2-1)
                                                                                 1273
  962 F2=TERPOL(Y1, Y2, Y3, TDIF2, T1, T2, T3)
                                                                                 1274
      IF(F2 .LT.0.00) F2=0.D0
                                                                                 1275
      IF(F2 .GT.1.D0) F2=1.D0
                                                                                 1276
C
                                                                                 1277
(
                                                                                 1278
                                                                                 1279
C
                                                                                 1280
```

```
GO TO 930
                                                                                  1281
  928 F2=1.00
                                                                                  1282
  930 IF(TDIF1.GE.CDF(J1.12)) GO TO 933
                                                                                  1283
  931 IF(TDIF1.GT.CDF(J1.L1)) GO TO 932
                                                                                  1284
      4.1=1.1-1
                                                                                  1285
      IF(1-L1)931.931.922
                                                                                  1286
C
                                                                                  1287
C
                                                                                  1288
C
                                                                                  1289
                                                                                  1290
  932 IF (L1 .NE. 1) GO TO 971
                                                                                  1291
      T1=CDF(J1.1)
                                                                                  1292
      T2=CDF(J1.2)
                                                                                  1293
      T3=CDF(J1..3)
                                                                                  1294
      Y1=0.C0
                                                                                  1295
      ¥2=0.0500
                                                                                  1296
      Y3=0.1500
                                                                                  1297
      GO TO 972
                                                                                  1298
  971 T1=CDF(J1.L1)
                                                                                  1299
      T2=CDF(J1.L1+1)
                                                                                  1300
      T3=CDF(J1.L1-1)
                                                                                  1301
      Y1=A(LI)
                                                                                  1302
      Y2=A(L1+1)
                                                                                  1303
      Y3=A(L1-1)
                                                                                 1304
  972 F1=TERPOL(Y1. Y2. Y3. TDIF1. T1. T2. T3)
                                                                                  1305
      IF(F1.LT.0.D0) F1=0.D0
                                                                                  1306
      IF(F1-GT-1-DO) F1=1-DO
                                                                                 1307
C
            .
                   .
                                                                                  1308
c
                                                                                  1300
C
                                                                                  1310
C
                                                                                  1311
      GO TO 940
                                                                                 1312
  933 F1=1.DO
                                                                                  1313
  940 FOT1=FCT1+F1+F2
                                                                                 1314
      GG TO 921
                                                                                  1315
  985 DENS(J)=FCT+0.10-1
                                                                                 1316
  990 CONTINUE
                                                                                  1317
      S(J1.2)=S(1.2)
                                                                                  1318
      CALL INVSTR(JI.DELTAT)
                                                                                 1319
      RETURN
                                                                                  1320
      END
                                                                                 1321
      FUNCTION TERPOL(X1.X2.X3.Y.Y1.Y2.Y3)
                                                                                 1322
      IMPLICIT REAL+8(A-H. 0-Z)
                                                                                 1323
      TERPOL=X1 + ((X2-X1)+(Y-Y1)/(Y2-Y1)) + ((X3-X2)/(Y3-Y2)-(X2-X1)/(Y3-Y2)
                                                                                  1324
     *(Y2-Y1))*((Y-Y1)*(Y-Y2)/(Y3-Y1))
                                                                                 1325
      RETURN
                                                                                 1326
      END
                                                                                  1327
      SUBROUTINE INVSTR(I.DELTAT)
                                                                                 1328
                                                                                  1329
      COF INVERSE STORAGE ROUTINE
C
                                                                                 1330
                                                                                 1331
      IMPLICIT REAL+8(A-H.O-R.T-Z)
                                                                                 1332
      COMMON /BLKA/S. COF. A.DENS.TI
                                                                                 1333
      INTEGER S
                                                                                 1334
      DIMENSION S(1000.4).CDF(1000.12).A(12).DENS(400).TI(2)
                                                                                 1335
      TEMPO=TI(1)
                                                                                 1336
      MINEL
                                                                                 1337
      DO 2263 IA=2.10
                                                                                 1338
      IF (CENS(IA).GE.1.DO) GG TO 2273
                                                                                 1339
      IF (DENS(1A).GT.0.DO) GC TO 2263
                                                                                 1340
      MIN=IA
                                                                                 1341
```

	TI(1)=TEMPO+DFLOAT(1A-1)+DELTAT	1342
2263	CONTINUE	1343
	MAX=11	1344
2273	GO TO 251 MAX=1A	1346
22.0	TI(2)=TEMPO+DFLOAT(IA-1)*DELTAT	1347
251	DENS(MIN)=0.00	1348
	DENS(MAX)=1.00	1349
	JA=0	1350
	DU 2283 KA=MIN.MAX	1351
	JA=JA+1	1352
	DENS(JA)=DENS(KA)	1353
2263	CONTINUE	1354
2267	DO 2293 LA=JA.11 DENS(LA)=1.DO	1356
	K=1	1357
	CDF(1.1)=TI(1)	1358
	CDF(1,12)=T1(2)	1359
	DC 260 M=2.6	1360
255	IF(A(M).GT.DENS(K).AND.A(M).LE.DENS(K+1)) GO TO 256	1361
	K=K+1	1362
	GC TO 255	1363
	Y=A(M)	1364
257	Y1=DENS(K) Y2=DENS(K+1)	1365 1366
	Y3=DENS(K+2)	1367
	TI=TI(I) + DFLOAT(K-1)+DELTAT	1368
	D1=Y2-Y1	1369
	D2=Y3-Y2	1370
	D3=Y3-Y1	1371
	DY1=Y-Y1	1372
	0Y2=Y-Y2	1373
	CTEMP =T1+(DY1+DELTAT/D1)+(DY1+DY2+DELTAT+(D1-D2)/(D1+D2+D3))	1374
	IF (CTEMP-LT-T1) CTEMP=T1	1375 1376
260	IF(CTEMP.GT.T1+2*DELTAT)CTEMP=T1+2*DELTAT CDF(I.M)=CTEMP	1377
200	DO 270 M=7.11	1378
265	IF(A(M).GT.DENS(K).AND.A(M).LE.DENS(K+1)) GO TO 266	1379
	K=K+1	1380
	GO TO 265	1381
266	Y=A(M)	1382
	Y1=DENS(K+1)	1383
	Y2=DENS(K)	1384
	Y3=DENS(K-1)	1385
	T1=TI(1) + OFLGAT(K)+DELTAT D1=Y2-Y1	1386
	D2=Y3-Y2	1388
	D3=Y3-Y1	1389
	041=4-41	1390
	DY2=Y-Y2	1391
	CTEMP =T1-(DY1+DELTAT/D1)-(DY1+CY2+DELTAT+(D1-D2)/(D1+D2+D3))	1392
	IF(CTEMP.LT.TI)CTEMP=T1	1393
	IF (CTEMP.GT.T1+2*DELTAT)CTEMP=T1+2*DELTAT	1394
270	CDF(I,M)=CTEMP	1395
	RETURN END	1396
	SUBROUTINE XDENS(MM.1.1500.J200)	1398
	IMPLICIT REAL+B(A-1-Q-R-T-Z)	1399
	COMMON /BLKB/TIME.	1400
	DIMENSION TIME(4).V(12)	1401
	GO TO (30.31.50.60).MM	1402

```
50
      CALL XRECT
                                                                                1403
      GG TO 70
                                                                                1404
60
      DO 61 M=1.12
                                                                                1405
61
      Y(M)=TIME(1)
                                                                                1466
      TIME(3)=0.00
                                                                                1407
      GC TO 70
                                                                                1408
   31 IF(TIME(4))32.32.33
                                                                                1409
32
      TIME(3)=(TIME(3)-TIME(1))/6.0
                                                                                1410
   33 TIME(1)=TIME(2)
                                                                               1411
34
      CALL NORMAL(J200)
                                                                                1412
      GC TC 70
                                                                               1413
30
      CALL XBETA(1500)
                                                                               1414
70
      CONTINUE
                                                                                1415
      RETURN
                                                                                1416
      END
                                                                                1417
      SUBRCUTINE XRECT
                                                                               1418
C
      SUBROUTINE TO CALCULATE THE INVERSE FUNCTION OF A RECTANGULAR DIST
                                                                                1419
C
      CALLING ARGUMENTS
                                                                               1420
         A IS THE MINIMUM VALUE OF THE VARIABLE
                                                                               1421
C
         B IS THE MAXIMUM VALUE OF THE VARIABLE
                                                                               1422
c
         Y IS THE ARRAY OF THE INVERSE FUNCTION OF THE DISTRIBUTION
                                                                               1423
C
      THE MEAN IS RETURNED IN A
                                                                               1424
C
      THE STANDARD DEVIATION IS RETURNED IN B
                                                                               1425
                                                                               1426
      IMPLICIT REAL+8(A-H.O-Z)
                                                                               1427
      COMMON /BLKB/TIME.Y
                                                                               1428
      DIMENSION TIME(4).Y(12)
                                                                               1429
      A=TIMF(1)
                                                                               1430
      B=TIME(3)
                                                                                1431
      DELTA=(TIME(3)-TIME(1))/10.00
                                                                               1432
      Y(1)=A
                                                                                1433
      Y( 12)=B
                                                                                1434
      Y(2)=Y(1)+DELTA+0.5D0
                                                                               1435
      DO 10 J=3.11
                                                                               1436
   10 Y(J)=Y(J-1)+DELTA
                                                                               E 437
      XMU=(A+B)/2.DO
                                                                               1438
      SIGMA=(B-A)/3.4641D0
                                                                               1439
      TIME(1)=XMU
                                                                               1440
      TIME (3) = SIGMA
                                                                                1441
       RETURN
                                                                               1442
      END
                                                                                1443
      SUBROUTINE XBETA(IG)
                                                                               1444
      IMPLICIT REAL+8(A-H.O-Z)
                                                                               1445
                                                                                1446
      COMMON /BLKB/TIME,Y
      DIMENSION TIME(4).Y(12)
                                                                                1447
      TERP (X1.X2.Y.Y1.Y2)=X1+((X2-X1)+(Y-Y1)/(Y2-Y1))
                                                                               1448
C TERP IS A ARITHMETIC STATEMENT FUNCTION TO DO LINEAR INTERPOLATION.
                                                                               1449
C IF X IS THE VALUE REQUIRED WHICH LIES BETWEEN THE VALUES OF X1 AND X2
                                                                               1450
C IN A TABLE AND Y1 AND Y2 ARE THECORRESPONDING VALUES WITH Y THE VALUE
                                                                               1451
C CORRESPONDING TO THE VALUE OF X REQUIRED.
                                                                               1452
C SUB PROGRAM TO CALCULATE THE CUMULATIVE INVERSE FUNCTION OF THE BETA
                                                                               1453
                             DENSITY FUNCTION
                                                                               1454
                                                                                1455
C
                                                                               1456
C
       CALLING ARGUMENTS.
                                                                               1457
         1. BETA # BETA OF THE DENSITY IF KNOWN. ENTER -1 IF IT IS
C
                                                                               1458
            DESIRED TO CALCULATE THESE POINTS BASED ON THE THREE PERT
                                                                               1459
C
C
            TIME ESTIMATES XSEE SUBROLTINE ALBET ..
                                                                                1460
C
         2. A # MINIMUM TIME ESTIMATE.
                                                                               1461
C
         3. XM # ALPHA OF THE DENSITY OR THE MCST LIKLEY TIME ESTIMATE.
                                                                               1462
         4. B . MAXIMUM TIME ESTIMATE.
                                                                               1463
```

C	5. IG # SEE GAMMA SUBROUTINE.	1464
C	6. Y # ARRAY 101 X 1 IN WHICH THE VALUES ARE CALCULATED	1465
C	7. SIGMA IS THE STANDARD DEVIATION OF THE DENSITY	1466
C	THE MEAN OF THE DENSITY IS RETURNED IN A. THE STANDARD DEVIATION OF THE DENSITY IS RETURNED IN TIMEX3<	1468
C	THE STANDARD DEVIATION OF THE DENSITY IS RETURNED IN TIMEAST	1469
-	A=TIME(1)	1470
	ALPHA=TIME(2)	1471
	B=TIME(3)	1472
	BETA=TIME(4)	1473
	Y(1)=A	1474
	Y(12)=8	1475
18	FALPHA=ALPHA+1.00	1476
	XM=ALPHA	1477
	FBETA=BETA + 1.00	1478
	SUM=ALPHA + BETA + 1.DO	1479
	FSLM= SUM + 1.00	1480
	SIGMA=(FALPHA+FBETA+(B-A)++2)/(((FSUM)++2)+(FSUM+1.D0))	1481
	SIGMA=DSQRT(SIGMA)	1482
	U = ((A*FBETA)+(G*FALPHA))/FSUM	1483
	CALL GAMMA (FALPHA, IG)	1484
	CALL GAMMA (FBETA.IG)	1485
	CALL GAMMA (FSUM.IG)	1486
	81=0.00	1488
	DELTAT=C/250D0	1489
	T3=0.00	1490
	FT=0.00	1491
	CONST=DLCG(FSUM)-DLOG(FALPHA)-DLOG(FBETA)-(SUM*DLOG(C))	1492
	Z3=0.D0	1493
	K=0	1494
	DO 50 J=2,20,2	1495
	D=DFLOAT(J-1)/20.0	1496
49	T1=T3	1497
	K=K+1	1498
	X=K	1499
	XX=K*2	1500
	AT=BT	1501
	T3=X+CELTAT	1502
	T2=(XX-1.D0)+T3/XX Z1=Z3	1503
	Z2=CChST+DLOG(4.D0)+(ALPHA+DLOG(T2))+(BETA+DLOG(C-T2))	1505
	Z2=DEXP(Z2)	1506
	Z3=CONST+(ALPHA*DLOG(T3))+(BETA*DLOG(C-T3))	1507
	Z3=DE XP (Z3)	1508
	FT=FT+Z1+Z2+Z3	1509
	BT=FT*DELTAT/6.DO	1510
	IF(BT.GT.1.DO) BT=1.DO	1511
	IF(D.GT.AT.AND.D.LE.BT) GO TO 51	1512
	GO TO 49	1513
51	NJ=(J/2)+1	1514
	Y(NJ)=TERP (T1.T3.D.AT.BT) + Y(1)	1515
50	CONTINUE	1516
	TIME(1)=U	1517
	TIME(2)=ALPHA	1518
	TIME (3)=SIGMA	
	TIME(4)=BETA RETURN	1520 1521
	# 보고 보 기하다 하는데 보다 되었다. 그는 사람들은 사람들은 사람들은 사람들은 사람들은 사람들은 사람들은 사람들은	1521
	SUBROUTINE GAMMA (X.M)	1523
c	The state of the s	1524

```
1525
  SUBROLTINE TO COMPUTE THE GAMMA FUNCTION XSINGLE PRECISIONS
                                                                                  1526
C
                                                                                  1527
C
         CALLING ANGUMENTS
                                                                                  1528
C
               X IS THE NUMBER OF WHICH THE GAMMA IS TO BE COMPUTED
c
               X IS A FLOATING NUMBER
                                                                                  1530
C
               M IS AN INTEGER NUMBER
                                                                                  1531
C
               M IS DEFINED AS FOLLOWS
                                                                                  1532
C
                    M MUST BE ZERO THE FIRST TIME THE SUBROUTINE IS CALLE
                                                                                  1533
C
      IMPLICIT REAL+8(A-H.O-R.T-Z)
                                                                                  1535
      DIMENSION Z(18)
                                                                                  1536
      IF(M) 1. 1. 2
                                                                                  1537
    1 B=1.DO
                                                                                  1538
      D=2.00
                                                                                  1539
      M=1
                                                                                  1540
      Z( 1)= 1.0000000000 00
      Z( 2)= .577215664D 00
Z( 3)=- .655878071D 00
                                                                                  1542
                                                                                  1543
      Z( 4)=- .C42002635D 00
                                                                                  1544
      Z( 5)= .166538611D 00
                                                                                  1545
      2( 6)=- .0421977340 00
                                                                                  1546
      Z( 7)=- .009621971D 00
                                                                                  1547
      2( 6)= .007218943D 00
                                                                                  1548
      Z( 9)=- .001165167D 00
                                                                                  1549
      Z(10)=- .000215241D 00
                                                                                  1550
      Z(11)= .000128050D 00
                                                                                  1551
      Z(12)=- .000020134D 00
                                                                                  1552
      Z(13)=- .000001250D 00
      Z(14)= .0000011330 00
                                                                                  1554
      Z(15)=- .000000205D 00
                                                                                  1555
      Z(16)= .000000006D 00
                                                                                  1556
      Z(17)= .000000005D 00
Z(18)=- .00000001D 00
                                                                                  1557
                                                                                  1558
    2 C=1.00
                                                                                  1559
    3 IF(X) 99.99. 4
                                                                                  1560
    4 IF(X-8) 7. 6. 5
                                                                                  1561
    5 IF (D-X) 8. 6. 9
                                                                                  1562
    6 G=1.00
                                                                                  1563
      GC TO 11
                                                                                  1564
    7 C=C+B/X
      X=X+B
                                                                                  1566
      GC TO 3
                                                                                  1567
      X= X-8
                                                                                  1568
                                                                                . 1569
      C=C+X
      GC TC 3
                                                                                  1570
    9 E=0.D0
                                                                                  1571
      DO 10 1=1.18
                                                                                  1572
      T=I
                                                                                  1573
   10 E=E+(Z(1)*(X**T))
                                                                                  1574
      G=B/E
                                                                                  1575
   11 X=C+G
                                                                                  1576
   99 RETURN
                                                                                  1577
      END
                                                                                  1578
      SUBROUTINE NORMAL(J2)
                                                                                  1579
    SUBROUTINE TO CALCULATE THE INVERSE FUNCTION OF THE NORMAL DISTRIB.
C
                                                                                  1580
   CALLING ARGUMENTS.
                                                                                  1581
     1. Y IS A FLOATING POINT ARRAY FOR DUTPUT OF THE CUMULATIVE INVERS
c
                                                                                  1582
         FUNCTION.
                                                                                  1583
C
     2. U IS THE MEAN OF THE DENSITY.
                                                                                  1584
     3. SIGMA IS THE STANDARD DEVIATION OF THE DENSITY.
                                                                                  1585
```

c	4.		1586
C	5.	그는 그 그는 그를 가는 그를 가는 것이 되었다. 그는 그는 그는 그는 그는 그는 그를 가는 그를 가는 그를 가는 것이 되었다.	1587
C		IS CALLED OR ANYTIME THE Z ARRAY IS USED AGAIN BY THE MAIN	1588
C		PROGRAM. WHEN JE IS MINUS THE Z TABLE IS NOT RECALCULATED	1589
C		THE PROGRAM RETURNS VALUES OF Y FOR THE FUNCTION	1590
C	-	# U 6Z * SIGMA FOR VALUES OF FXY<#.0102991.0.	1591
C	The second	< WILL CONTAIN U - 3 STANDARD DEVIATIONS. IF & DR.O IF NEGATIVE	1592
C	-	UES OF Y ARE NOT CALCULATED. YX 224 WILL CONTAIN U & 3 STANDARD	1593
C		LATIONS.	1594
		MPLICIT REAL*8(A-H.O-Z)	1595
	-	OMMON /BLKB/TIME.Y	1596
		IMENSIGN TIME(4).Y(12)	1597
		IMENSION Z(10)	1598
		= TI ME (2)	1599
		IGMA=TIME(3)	1600
		F (J2) 5,10,10	1601
		2=-1	1602
		(1)=-1.6449300	1603
		(2)=-1.0364500	1604
	_	(3)=-0.6744900	1605
	100	(4)=-0.3853200	1606
	-	(5)=-0.12566D0	1607
		(6)=0.1256600	1608
		(7)=0.3853200	1609
		(8)=0.6744900	1610
	_	(9)=1.0364500	1611
		(10)=1.6449300	1612
5		ZERO=U-(3.DO*SIGMA)	1613
		F (TZERO) 1,2,2	1614
		(1)=0.00	1615
		0 10 8	1616
-		(1)=TZERO	1617
		0 9 1=2.11	1618
		(1)=Z(1-1)*SIGMA+U	1619
		F (Y(1)) 6,6,9	1620
		(1)=0.00	1621
9	-	ONTINUE	1622
		(12)=U+3.D0+SIGMA	1623
		ETURN	1624
	_	ND	1625
		UBROUTINE RD(J+A+N)	1626
		IMENSION A(N)	1627
		EAD(J) A	1628
		ETURN	1629
	E	ND .	1630

6. DECOMP

C	DECOMPOSITION PROGRAM	1
c	DECCMP IDENTIFIES PARALLEL SUBNETWORKS AND SERIES SUBNETWORKS	2
c	SECOND IDENTIFIES PRINCEES SOURCE SOURCE SOURCE SOURCE	4
C		5
C	THE FOLLOWING IS AN ALPHABETICAL LISTING OF THE VARIABLES AND	6
c	ARRAYS THAT ARE USED IN THIS MAIN PROGRAM AND ITS SUBROUTINES	7
c		9
c	ARCS = THE NUMBER OF ARCS IN THE SUBNETWORK	10
C	BDNUM(I) = THE BUNDLE NUMBER TO WHICH NODE I IS ASSIGNED	11
C	CARC = AN ARRAY PUNCHED FOR USE IN PROGRAM: SYNTHESIS	12
C	CHECK = ARRAY USED TO STORE ARCS HAVING THE SOURCE AND THE SINK AS THEIR ONLY NODES	13
c	CINSUB = THE TEMPORARY NUMBER OF SUBNETWORKS FOUND IN THE	15
c	PREVIOUS STEP	16
C	HEAD(I)=TERMINATING NODE FOR ARC I IN THE RENUMBERED SUBNETWORK	17
C	INSTN = THE CURRENT INSTRUCTION NUMBER. PUNCHED FOR SYNTHESIS LNCDEN = THE LARGEST NOCE NUMBER BEING READ IN	18
c	MAXND = THE LARGEST NODE NUMBER THAT HAS ALREADY BEEN ASSIGNED	20
c	AT LEAST TEMPORARILY TO A BUNDLE	21
C	NARCSS(1) = THE NUMBER OF ARCS IN SUBNETWORK I	22
C	NODES=TEMPURARY ARRAY USED TO ORDER THE NCDE NUMBERS IN A SUB	23
C	NSUB = THE TOTAL OF SUBNETWORKS THUS FAR NTARC = THE NUMBER OF ARCS IN THIS SUBNETWORK	24 25
c	NUMBD = THE NUMBER OF BUNDLES CREATED	26
c	NUMSUB=NUMBER OF MINIMUM SUBNETWORKS IDENTIFIED	27
C	PPOST = ARRAY OF PAST POSTS USED IN CUT SUBROUTINE	28
C	S(I) = THE STARTING NODE FOR ARC I	29 30
c	SINK = THE NODE NUMBER CORRESPONDING TO THE SINK SINKS(I) = THE SINK IN SUBNETWORK I	31
c	SOURC(I) = SOURCE NODE IN SUBNETWORK I	32
C	SCURCE = THE NODE NUMBER CORRESPONDING TO THE SOURCE	33
c	STEP = STAGE NUMBER	34
c	SUBNET(1,J) = THE LTH ARC IN THE JTH SUBNETWORK SUMANC = THE CURRENT NUMBER OF ARCS IN SUBNETWORK NSUB	35 36
č	T(1) = THE TERMINATING NODE FOR ARC I	37
C	TAIL(1)=STARTING NODE FOR ARC 1 IN THE RENUMBERED SUBNETWORK	38
C	TARC = THE SUBNETWORK WITHOUT THE ARCS INVOLVING NODE K	39
c	TLNCEN = TEMPORARY LARGEST NODE NUMBER INSUB = THE NUMBER OF SUBNETWORKS CREATED IN THE CURRENT STAGE	40
c	TSASUB = THE TEMPURARY SUBNETWORK BEING USED IN STAGE	42
c	SUBROUTINE	43
C	TSUBN = THE NUMBER OF THE SUBNETWORK CURRENTLY BEING CONSIDERED	44
c	TYPESN = THE TYPE OF SUBNETWORK BEING CONSIDERED:	45
c	1 = BUNDLE SUBNETWORK 2 = CUT SUBNETWORK	46
c	A NON-DECOMPOSABLE SUBNETWORK IS ONE THAT IS NOT COMPOSED OF	48
c	SMALLER SUBNETWORKS WHICH ARE CONNECTED IN EITHER SERIES OF	49
C	PARALLEL	50
c		51 52
c	FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS. LET	53
č	MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL	54
C	PROJECT NETWORK	55
C	MSMAX = THE MAXIMUM NUMBER OF ACTIVITIES ALLOWED IN ANY ONE	56
c	SUBNETWORK NSUBMAX = THE MAXIMUM NUMBER OF SUBNETWORKS ALLOWED IN THE	57 58
c	DECUMPUSITION PROCESS	59
c	ACINSMAX = THE MAXIMUM NUMBER OF INSTRUCTIONS GENERATED BY THE	60

```
DECOMPOSITION PROCESS
         CURRENTLY, MMAX=1000; MSMAX=500; NSUBMAX=100; NDINSMAX=100
                                                                                 62
                                                                                 63
      DIMENSION S(MMAX), T(MMAX), SUBNET(MMAX, NSUBMAX), SOURC (NSUBMAX),
                                                                                 64
C
c
                NARCSS(NSUBMAX).CARD(NOINS.27)
                                                                                 65
C
                                                                                 66
C
         IN SUBROUTINE NETIN THE ARRAY DIMENSIONS ARE:
                                                                                 67
C
     DIMENSION FILE3(3*MMAX+5).IORG(MMAX).ITERM(MMAX).NUMS(MMAX)
                                                                                 68
      EQUIVLAENCE (FILE3(MMAX+6).IDRG(1)).(FILE3(2*MMAX+6).ITERM(1))
                                                                                 69
C
                                                                                 70
         IN SUBROUTINE BUNDLE THE ARRAY DIMENSIONS ARE:
                                                                                71
C
     DIMENSION BONUM(MMAX) . CHECK(MMAX)
                                                                                 72
C
                                                                                 73
         IN SUBROUTINE CUT THE ARRAY DIMENSIONS ARE:
                                                                                 74
C
c
      DIMENSION ORIGIN(MMAX).POST(MMAX).TARC(MMAX).RCUT(NSUBMAX).
                                                                                 75
                PPOST (MMAX)
                                                                                 76
C
C
                                                                                 77
         IN SUBROUTINE PRESUB THE ARRAY DIMENSIONS ARE:
                                                                                 78
C
C
     DIMENSION HEAD(MMAX).TAIL(MMAX).NODE(MMAX).FILE3(3*MSMAX+3).
C
                NET (3. MSMAX)
                                                                                 80
C
                                                                                 81
      IMPLICIT INTEGER+2 (A-Z)
                                                                                 82
      CCMMON SUBNET.S.T.SOURC.SINKS.NARCSS.NSUB.TNSUB
                                                                                 83
      COMMON /NEW/ CARD. INSTN
                                                                                 84
      COMMON/PREP/ NUMSUB
                                                                                 85
                                 F3/11/+
                                              F5/13/
                                                                                 86
      INTEGER CARULIDO.27)
                                                                                 87
      INTEGER+2 S(1000).T(1000).SUBNET(1000.100).SOURC(100).SINKS(100).
                                                                                 88
                                                                                 89
     *NAFCSS(100)
      WRITE(6,100)
                                                                                 90
     FORMAT(1H1.132(***)/1x.132(***)/*OTHIS IS THE CUTPUT FROM THE DECO
     *MPOSITION PROGRAM: DECOMP*/1H0.132(***)/1X.132(***))
                                                                                 92
                                                                                 93
      REGING F3
      READ(F3)
                                                                                 94
      REWIND F5
                                                                                 95
      READ(F5)
                                                                                 97
C
C
         INITIALIZE INSTN
                                                                                 58
         INSTN = THE CURRENT INSTRUCTION NUMBER
C
                                                                                 99
C
                                                                                100
6000 INSTN=0
                                                                                101
C
                                                                                102
         INITIALIZE CARD
C
                                                                                103
C
                                                                                104
      DC 22200 I=1.100
                                                                                105
      DG 22200 J=1.27
                                                                                106
22200 CARC(I.J)=0
                                                                                107
                                                                                108
C
         READ THE NETWORK IN
                                                                                109
C
                                                                                110
      CALL NETIN (LNODEN)
                                                                                111
      REWING F3
                                                                                112
      READ(F3)
                                                                                113
      STEP=1
                                                                                114
      TYPESN=1
                                                                                115
                                                                                116
      TSUBN=1
      NUMSUB=0
                                                                                117
      CALL BUNDLE (LNODEN.TSUBN)
                                                                                118
C
                                                                                119
C
         PRINT OUT THIS STAGE OF THE BREAKUP
                                                                                120
                                                                                121
C
```

	L STAGE (TYPESN.TSUBN)
GC	TC 70
	FIND THE NUMBER OF NEWLY CREATED SUBNETWORKS
CI	SUB=0
TSU	JBN=NSUB-LOOP
TYF	PESN=1
00	20 I=1.LOOP
	FIND THE NEXT SUBNETWORK TO BE FURTHER SUBDIVIDED
TSU	BN=TSUBN+1
	FINC THE LARGEST NODE NUMBER IN SUBNETWORK TSUBN
CAL	L NCDER (TLNDEN.TSUBN)
	FIND THE BUNDLE SUBNETWORKS
CAL	L BUNCLE (TLNDEN.TSUBN)
	PRINT OUT THIS STAGE OF THE BREAKUP
CAL	L STAGE (TYPESN.TSUBN)
	IF THERE IS ONLY ONE BUNDLE FOUND IN SUBNETWORK TSUBN. WE ARE
	FINISHED. PRINT OUT ITS COMPONENT ARCS
IF	(TNSUB.EQ.1) CALL ENDSNT (TSUBN)
-	
	COUNT THE NEW NUMBER OF SUBNETWORKS CREATED
CT	ISUE=TNSUB+CTNSUB
• • •	
	IF ALL SUBNETWORKS ARE IN THEIR SMALLEST FORM, WE ARE FINISHED
IF	(CTNSUB.EQ.0) GO TO 90
	TO 70
100	P=CTNSLB
	ISU8=0
	JBN=NSUB-LOOP
	FIND THE NUMBER OF NEWLY CREATED SUBNETWORKS
TY	PESN=2
	30 I=1.LOOP
-	
	FINC THE NEXT SUBNETWORK TO BE FURTHER SUBDIVIDED
	THE THE HEAT SUBMETHORN TO BE TORTHER SUBSTITUTED
16	JBN=TSUBN+1
31.50	(TSUBN.EQ.1) GO TO 55
	113000000000000000000000000000000000000
	FIND THE LARGEST NODE NUMBER IN SUBNETWORK TSUBN
	FIND THE ENGOST MODE NOMBER IN SUBNETHURK 1500M
	A MODER ATLANCEN TOURNS
	L NODER (TLNDEN.TSUBN)
	THE THE CUT CURNETHORNE
	FIND THE CUT SUBNETWORKS

```
163
      CALL CUT (TSUBN.TLNDEN)
                                                                                184
C
         PRINT CUT THIS STAGE OF THE BREAKUP
                                                                                185
C
                                                                                186
      CALL STAGE (TYPESN.TSUBN)
                                                                                187
C
                                                                                188
         IF THERE ARE NO CUIS FOUND IN SUBNETWORK TSUBN. WE ARE FINISHED
                                                                                189
C
c
         PRINT OUT ITS COMPONENT ARCS
                                                                                190
                                                                                191
C
      IF (TNSUB.EQ.1) CALL ENDSNT (TSUBN)
                                                                                192
C
                                                                                193
         COUNT THE NEW NUMBER OF SUBNETWORKS CREATED
                                                                                194
C
c
                                                                                195
                                                                                196
30
      CINSUE=TNSUB+CINSUB
                                                                                197
C
         IF ALL SUBNETWORKS ARE IN THEIR SMALLEST FORM. WE ARE FINISHED
                                                                                198
C
C
                                                                                199
      IF (CTNSUB.EU.O) GO TO 90
                                                                                200
                                                                                201
      GO TO 70
85
      LCOP=CINSUB
                                                                                202
      GO TU 10
                                                                                203
70
      WRITE (6.900) STEP
                                                                                204
      FORMAT (1H1.5X. STAGE . 13. BREAKUP )
900
                                                                                205
      STEP=STEP+1
                                                                                206
C
                                                                                207
         LET'S GO BACK TO THE APPROPRIATE LOOP FOR THE NEXT STAGE
                                                                                208
C
C
                                                                                209
      GO TO (80.85). TYPESN
                                                                                210
90
      CCNTINUE
                                                                                211
C
                                                                                212
         STORE INFORMATION FOR SYNTHESIS PROGRAM
                                                                                213
C
                                                                                214
C
                                                                                215
      CALL PUNSYN
      WEITE(6,7001) NUMSUB
                                                                                216
                             THE NUMBER OF NON-CECOMPOSABLE SUBNETWORKS I
7001 FORMAT(IHO.
                                                                                217
     +S *(3)
                                                                                218
      WRITE (6.9000)
                                                                                219
9000 FORMAT (1H1)
                                                                                220
      STOP
                                                                                221
      END
                                                                                222
      SUBROUTINE NETIN (LNCDEN)
                                                                                223
      IMPLICIT INTEGER#2 (A-Z)
                                                                                224
      COMMON SUBNET.S.T.SOURC.SINKS.NARCSS.NSUB.TNSUB
                                                                                225
      INTEGER*2 S(1000).T(1000).SUBNET(1000.100).SOURC(100).SINKS(100).
                                                                                226
     *NARCSS(100)
                                                                                227
      INTEGER#4 FILE3(3005)
                                                                                228
      INTEGER#4 | IURG(1000) . ITERM(1000) . NUMS(1000)
                                                                                229
      EQUIVALENCE (FILE3(6).NUMS(1)).(FILE3(1006).IORG(1)).
                                                                                230
     *(FILE3(2006).ITERM(1))
                                                                                231
     INTEGER
                                  F3/11/
                                                                                232
      READ(F3) FILE3
                                                                                233
C
                                                                                234
C
         ZERGIZE SUURC ARRAY
                                                                                235
c
                                                                                236
      DO 20 1=1.50
                                                                                237
20
      SOURC(1)=0
                                                                                238
C
                                                                                239
         READ IN THE INITIAL NETWORK LIMITS
C
                                                                                240
                                                                                241
      ARCS = FILE3(1)
                                                                                242
      SOURCE = FILE3(3)
                                                                                243
```

```
SINK = FILE3(4)
                                                                                  244
      LACDEN = FILE3(5)
                                                                                  245
      WRITE (6.200)
                                                                                  246
200
      FCRMAT (1HO.5X. INPUT STAGE )
                                                                                  247
      WRITE (6.210) ARCS. SUURCE. SINK. LNODEN
                                                                                  248
      FORMAT(1H0./6x. THE SIMPLIFIED NETWORK HAS . 12x.14. ARCS . . . //6x.
210
                                                                                  249
     * THE SOURCE IS NODE NUMBER . 11X. 13. //6X. THE SINK IS NODE NUMBER .
                                                                                  250
     +13x.13.//6x. 'THE LARGEST NODE IS NODE NUMBER .5x.13)
                                                                                  251
      BRITE (6,220)
                                                                                  252
      FGRMAT(IHO.5X. THE SIMPLIFIED NETWORK AS READ IN IS: *.///6X.
220
                                                                                  253
     **ARC NUMBER . 5x. ORIGIN NODE . 5x. TERMINAL NODE !)
                                                                                  254
                                                                                  255
         READ IN EACH ARC AND ITS STARTING AND TERMINATING NODES
C
                                                                                  256
         S = THE NODE NUMBER FOR THE START OF AN ARC
C
                                                                                  257
C
         T = THE TERMINAL NODE OF AN ARC
                                                                                  258
C
                                                                                  259
      DO 10 J=1.ARCS
                                                                                  260
      I=NUMS(J)
                                                                                  261
      S(I)=ICRG(J)
                                                                                  262
      T(I)=ITERM(J)
                                                                                  263
      WRITE(6.240)1.5(1).T(1)
                                                                                  264
240
      FORMAT (1H .8x.13.13x.13.14x.13)
                                                                                  265
C
                                                                                  266
C
         CREATE THE FIRST SUBNETWORK
                                                                                  267
C
                                                                                  268
10
      SCHNET(J.1)=1
                                                                                  269
      SOURC(1)=SOURCE
                                                                                  270
                                                                                  271
      SINKS(1)=SINK
      NARCSS(1)=ARCS
      NSLB=1
                                                                                  273
      RETURN
                                                                                  274
      END
                                                                                  275
      SUBROUTINE NODER (TLNDEN.TSUBN)
                                                                                  276
C
                                                                                  277
         FINDS LARGEST NODE NUMBER IN THE SUBNETWORK TSUBN
C
                                                                                  278
C
      IMPLICIT INTEGER#2 (A-Z)
                                                                                  280
      COMMON SUBNET, S. T. SOURC, SINKS, NARCSS, NSUB, TNSUB
                                                                                  281
      INTEGER*2 S(1000).T(1000).SUBNET(1000.100).SCURC(100).SINKS(100).
                                                                                  282
     *NARCSS(100)
                                                                                  283
      ARCS=NARCSS(TSUBN)
      TLNDEN=0
                                                                                  285
      DO 20 J=1.ARCS
                                                                                  286
      A=SUBNET(J.TSUBN)
                                                                                  287
      M=S(A)
                                                                                  288
      N=T(A)
                                                                                  289
      MAXND=N
                                                                                  290
      IF (M.GT.N) MAXND=M
      IF (MAXND.GT.TLNDEN) TLNDEN=MAXND
                                                                                  292
      CCNTINUE
20
      RETURN
                                                                                  294
                                                                                  295
      END
      SUBROUTINE ENDSNT (TSUBN)
                                                                                  296
                                                                                  297
C
C
         PRINTS SMALLEST BREAKDOWN OF SUBNETWORK TSUBN
                                                                                  298
C
                                                                                  299
      IMPLICIT INTEGER+2 (A-Z)
                                                                                  300
      COMMON SUBNET. S. T. SUURC. SINKS. NARCSS. NSUB. TNSUB
                                                                                  301
      INTEGER+2 S(1000).T(1000).SUBNET(1000.100).SOURC(100).SINKS(100).
                                                                                  302
     *NARC55(100)
                                                                                  303
      SOURCE=SOURC(TSUBN)
                                                                                  304
```

```
SINK=SINKS(TSUBN)
                                                                                 305
      WRITE (6.100) TSUBN
                                                                                 306
      FORMAT (1H0.//16x, SUBNETWORK . 13. IS A NON-DECOMPOSABLE NETWORK
                                                                                 307
100
     * . . / 16x . ' IT IS CUMPOSED CF: ')
                                                                                 308
      WRITE (6.200) SOURCE.SINK
                                                                                 309
      FCRMAT (1H0.19X. SOURCE NODE = 1.13.//20X. SINK NODE
200
                                                                                 310
      M=NARCSS(TSUBN)
                                                                                 311
      MFITE (6,500)
                                                                                 312
500
      FORMAT(1HO.15X. THE SUBNETWORK AS IT IS READ IN: 1)
                                                                                 313
      WRITE (6.400)
                                                                                 314
      FCRMAT (1H0.19x.'ARC'.2x.'S(ARC)'.2x.'T(ARC)')
                                                                                 315
400
      DO 10 1=1.M
                                                                                 316
      N=SUBNET(1.TSUBN)
                                                                                 317
10
      WRITE (6.300) N. S(N) . T(N)
                                                                                 318
      FORMAT (1H .19x.13.3x.13.5x.13)
                                                                                 319
300
C
                                                                                 320
         SET THSUB-O SO THAT THE REMAINING NUMBER OF SUBNETWORKS DOESN'T
                                                                                 321
C
         INCLUDE THIS MINIMUM SUBNETWORK
                                                                                 322
C
                                                                                 323
C
      TNSUB=0
                                                                                 324
C
                                                                                 325
         PREPARE THE MINIMUM SUBNETWORK FOR SUBNETWORK ANALYSIS
                                                                                 326
C
                                                                                 327
C
      CALL PRESUB(TSUBN)
                                                                                 328
      RETURN
                                                                                 329
      END
                                                                                 330
      SUBROUTINE STAGE (TYPESN. TSUBN)
                                                                                 331
C
                                                                                 332
         PRINTS OUT THE CURRENT STAGE OF BREAKUP
                                                                                 333
C
C
                                                                                 334
      IMPLICIT INTEGER#2 (A-Z)
                                                                                 335
      COMMON SUBNET.S.T.SOURC.SINKS.NARCSS.NSUB.TNSUB
                                                                                 336
      INTEGER*2 S(1000).T(1000).SUBNET(1000.100).SOURC(100).SINKS(100).
                                                                                 337
     *NAFCSS(100)
                                                                                 338
      COMMON /NEW/ CARD, INSTN
                                                                                 339
      INTEGER CARD(100.27)
                                                                                 340
C
                                                                                 341
         THSUB=THE NUMBER OF NEW SUBNETWORKS RESULTING FROM THE BREAKUP
C
                                                                                 342
C
         IN THIS STAGE
                                                                                 343
C
                                                                                 344
         TNSUB=1 IMPLIES NO BREAKUP OCCURRED IN THIS STAGE
                                                                                 345
C
                                                                                 346
      IF (TNSUB.EQ.1) GO TC 600
                                                                                 347
      WRITE (6.100) TSUBN
                                                                                 348
      FCRMAT (1H0,/6x, SUBNETWORK .. 13, IS COMPOSED OF SUBNETWORKS: .)
100
                                                                                 749
      ISNSUB=NSUB-THSUB
                                                                                 350
      M=TSASUB+1
                                                                                 351
      WRITE (6.300)(1.1=M.NSUB)
                                                                                 352
300
      FORMAT (40.5x.13.20( ... 13))
                                                                                 353
                                                                                 354
C
         PREPARE INFORMATION FOR SYNTHESIS PROGRAM
C
                                                                                 355
C
                                                                                 356
      INSTN=INSTN+1
                                                                                 357
      ISORP=0
                                                                                 358
      IF (TYPESN.EG.1) ISORP=1
                                                                                 359
      CARD(INSTN. 1)=INSTN
                                                                                 360
      CARD(INSTN.2)=TSUBN
                                                                                 361
      CARD(INSTN. J)=ISORP
                                                                                 362
      DG 2250 I=M. NSUB
                                                                                 363
      K=1-M+4
                                                                                 364
 2250 CARD(INSTN.K)=1
                                                                                 365
```

```
IF (TYPESN-EQ-1) GO TG 60
                                                                                  366
      WRITE (6.400)
                                                                                  367
400
      FORMAT (1HO.5X. 'IN SERIES')
                                                                                  368
      GO TO 600
                                                                                  369
60
      BRITE (6.500)
                                                                                  370
      FORMAT (1HO.5X. 'IN PARALLEL')
500
                                                                                  371
600
      RETURN
                                                                                  372
      END
                                                                                  373
      SUBROUTINE BUNDLE (LNODEN.TSUBN)
                                                                                  374
C
                                                                                  375
C
         BUNDLE IDENTIFIES PARALLEL SUBNETWORKS CONNECTING DESIGNATED
                                                                                  376
C
         SOURCE AND SINK
                                                                                  377
C
                                                                                  378
      IMPLICIT INTEGER#2 (A-Z)
                                                                                  379
      CCMMON SUBNET.S.T.SOURC.SINKS.NARCSS.NSUB.TNSUB
                                                                                  380
      INTEGER+2 S(1000). T(1000). SUBNET(1000.100). SCURC(100). SINKS(100).
                                                                                  381
     ONARCSS(100)
                                                                                  382
      INTEGER+2 BDNUM(1000) . CHECK(1000)
                                                                                  383
C
                                                                                  384
C
         GROUP NODES INTO BUNDLES
                                                                                  385
C
                                                                                  386
       SCURCE=SCURC(TSUBN)
                                                                                  387
       SINK=SINKS(TSUBN)
                                                                                  388
      NUMBD=1
                                                                                  389
C
                                                                                  350
C
         ZEROIZE THE BONUM ARRAY
                                                                                  391
C
                                                                                  392
      DO 10 I=1.LNODEN
                                                                                  393
   10 BDNUM(1)=0
                                                                                  394
      K=SUBNET(1.TSUBN)
                                                                                  395
      M=S(K)
                                                                                  396
      N=T(K)
                                                                                  397
      BONUM (M) =1
                                                                                  398
      BONUM(N)=1
                                                                                  399
      MAXND=N
                                                                                  400
      IF (M.GT.N) MAXND=M
                                                                                  401
      ARCS=NARCSS(TSUBN)
                                                                                  402
      IF (ARCS.EQ.1) GO TO 515
                                                                                  403
      DG 1 K=2.ARCS
                                                                                  404
      I=SUBNET(K.TSUBN)
                                                                                  405
      BDNUM (SOURCE) =0
                                                                                  406
                                                                                  407
      BONUM(SINK)=0
      M=5(1)
                                                                                  408
      N=T(I)
                                                                                  409
      IF (M.GT.MAXND) MAXND=M
                                                                                  410
      IF (N.GT.MAXND) MAXND=N
                                                                                  411
C
                                                                                  412
         IF AT LEAST I NODE ON THE ARC HAS NOT BEEN ASSIGNED TO A BUNDLE
C
                                                                                  413
         GO TO 2
                                                                                  414
C
C
                                                                                  415
      IF (BONUM(M).EQ.O) GO TO 2
                                                                                  416
C
                                                                                  417
         IF CNLY THE TERMINAL NODE ON THE ARC HAS NOT BEEN ASSIGNED TO A
C
                                                                                  418
C
         BUNDLE. GO TO 3
                                                                                  419
C
                                                                                  420
      IF(BCNUM(N).EQ.O) GO TO 3
                                                                                  421
C
                                                                                  422
         IF BOTH NUDES ON THE ARC HAVE BEEN ASSIGNED TO THE SAME BUNDLE
                                                                                  423
C
C
         EVERYTHING IS OKAY. GO TRY ANOTHER ARC
                                                                                  424
C
                                                                                  425
      IF (BCNUM(N).EQ.BDNUM(M)) GO TO 1
                                                                                  426
```

```
427
c
         IF THE NUDES ON THE ARC ARE ASSIGNED TO DIFFERENT BUNDLES.
                                                                                428
         THEN THESE TWO BUNDLES SHOULD BE POOLED
c
                                                                                429
c
                                                                                430
      IF(BONUM(N).LT.BONUM(M)) GO TO 6
                                                                                431
c
                                                                                432
c
         POOL BUNDLES
                                                                                433
c
         THE BUNDLE WITH THE LARGER BUNDLE NUMBER IS POOLED INTO THE
                                                                                434
C
         BUNDLE WITH THE SMALLER BUNDLE NUMBER
                                                                                435
c
         THE BUNDLE NUMBERS OF ALL BUNDLES ARE ALL ADJUSTED
                                                                                436
C
                                                                                437
      MAXBD=EDNUM(N)
                                                                                438
      MINBD=BDNUM(M)
                                                                                439
      GC TO 7
                                                                                440
    6 MAXBD=BDNUM(M)
                                                                                441
      MINBO=60NUM(N)
                                                                                442
    7 DC 5 J=1.MAXND
                                                                                443
      B=BDNUM(J)
                                                                                444
      IF (B.EQ.MAXBD) BDNUM(J)=MINBD
                                                                                445
      IF (B.GT.MAXBD) BDNUM(J)=BDNUM(J)-1
                                                                                446
    5 CENTINUE
                                                                                447
      NUMED=NUMBD-1
                                                                                448
      GO TO 1
                                                                                449
C
                                                                                450
c
         IF BOTH NODES ON THE ARC ARE UNASSIGNED. GO TO 4 WHERE A NEW
                                                                                451
c
         BUNDLE IS CREATED
                                                                                452
c
                                                                                453
    2 IF(BDNUM(N).EQ.0) GO TO 4
                                                                                454
C
                                                                                455
         ASSIGN THE ORIGIN NODE OF THE ARC TO THE BUNDLE CONTAINING THE
C
                                                                                456
C
         TERMINAL NODE
                                                                                457
C
                                                                                45A
      BDNUM(M)=BDNUM(N)
                                                                                459
      GG TG 1
                                                                                460
C
                                                                                461
         ASSIGN THE TERMINAL NODE OF THE ARC TO THE BUNDLE CONTAINING
C
                                                                                462
C
         THE DRIGIN NOVE OF THE ARC
                                                                                463
C
                                                                                464
    3 BDNUM(N)=BDNUM(M)
                                                                                465
      GO TO 1
                                                                                466
C
                                                                                467
C
         CREATE A NEW BUNDLE
                                                                                468
C
                                                                                469
    4 NUMBD=NUMBD+1
                                                                                470
      BONUM (MI=NUMBO
                                                                                471
      BDNUM(N)=NUMBD
                                                                                472
    1 CONTINUE
                                                                                473
515
      CONTINUE
                                                                                474
      BCNUM(SINK)=0
                                                                                475
C
                                                                                476
         IF WE ONLY HAVE I BUNDLE FROM THE SUBNETWORK. WE ARE FINISHED
C
                                                                                477
C
                                                                                478
      IF (NUMBD.EQ.1) GO TO 219
                                                                                479
C
                                                                                480
         ZERGIZE CHECK ARRAY
                                                                                481
C
C
                                                                                482
      DC 290 1=1.ARCS
                                                                                483
290
      CHECK(1)=0
                                                                                484
      L=0
                                                                                485
                                                                                486
C
         THE NODES ARE IN BUNDLES. PUT THE ASSOCIATED ARCS INTO
C
                                                                                487
```

	APPROPRIATE PARALLEL SUBNETWORKS	488
CO	33 I=1.NUMBD	490
-	MARC=0	491
	UE=NSUB+1	492
00	34 K=1.ARCS	493
M=	SUBNET(K. TSUBN)	494
N=	S(M)	495
		496
	SCURCE AND SINK HAVE BUNDLE NUMBER 0	497
		498
LF	(N.EQ.SOURCE) N=T(M)	499
	(BCNUM(N).EQ.I) GG TO 239	500
	(N-EQ-SINK) GO TO 229	501
GC	TC 34	502
	Tripost sala and animals animals and animals animals and animals animals and animals and animals and animals animals and animals animals and animals animals animals and animals a	503
	SPECIAL CASE: BUNDLE HAS ONLY 2 NCDES: SOURCE, SINK.	504
	PUT ALL ARCS THAT ARE PARALLEL SUBNETWORKS BY THEMSELVES INTO	509
	THE CHECK ARRAY	500
		50
	291 J=1,K	500
	CHECK(J)	509
	(w.EQ.O) GO TO 292	51
	(M.EQ.N) GO TO 34	51
	NT I NUE	512
	ECK(J)=M	51
GO	10 34	514
		51
	THIS ARC IS IN THE BUNDLE I. HENCE IT IS IN THE ITH NEW	510
	SUBNETWORK	51
		51
-	MARC=SUMARC+1	519
	BNET (SUMARC. NSUB) = M	52
CO	NT INUE	52
		52
	CREATE NEW SUBNETWORKS	52
		52
	TO THE CHARLE HE HE HELD OF THE MAN AND CHARLEST THE	52
	IF THIS BUNDLE HAS NO NODES, PUT AN ARC SUBNETWORK INTO	520
	SUBNET (1.NSUB)	52
	ASSUMED SO AL SO TO TO	52
	(SUMARC.EQ.O) GO TO 333	53
	RCSS(NSUB)=SUMARC	53
	NKS(NSUB)=SINK	53
	10 33	53
00	10 33	53
	STORE THE SUBNETWORKS THAT HAVE ONLY SOURCE AND SINK NODES	53
	STORE THE SUBHETHORNS THAT HAVE UNET SOURCE AND STAK HODES	53
511	MARC=1	53
-	L+1	538
	BNET([.NSUB)=CHECK(L)	539
	TO 343	540
	NT INVE	54
	SUB=NUMBD	54
	THEM	54:
		54
EN		546
EN	BROUTINE CUT (TSUBN.LNODEN)	
EN		545 546

```
c
                                                                                 549
         CUT ALSO IDENTIFIES THE CUT GROUPS: THAT IS. THE SUBNETWORKS
                                                                                 550
c
         WHICH ARE IN SERIES AND CONNECTED BY THE CUT NODES
                                                                                 551
C
                                                                                 552
                                                                                 553
      IMPLICIT INTEGER+2 (A-Z)
      COMMON SUBNET. S. T. SOURC. SINKS . NARCSS . NSUB. TNSUB
                                                                                 554
      INTEGER+2 S(1000).T(1000).SUBNET(1000.100).SCURC(100).SINKS(100).
                                                                                 555
     *NARCSS(100)
                                                                                 556
      INTEGER*2 TARC(1000).GRIGIN(1000).PDST(1000).RCUT(100).PPOST(1000)
                                                                                 557
                                                                                 558
C
                                                                                 559
C
         FIND THE CUT NODES
C
         NCUT IS THE NUMBER OF CUT NODES FOUND THUS FAR
                                                                                 560
C
                                                                                 561
      NCUT=C
                                                                                 562
      ARCS=NARCSS(TSUBN)
                                                                                 563
      SCURCE=SOURC(TSUBN)
                                                                                 564
      SINK=SINKS(TSUBN)
                                                                                 565
C
                                                                                 566
         THE DO LOOP DOWN TO STATEMENT NUMBER 1 DETERMINES THE CUT NODES
C
                                                                                 567
C
                                                                                 568
      DC 1 K=1.LNODEN
                                                                                 569
C
                                                                                 570
C
         CHECK TO SEE IF NODE K IS ACTUALLY IN THE SUBNETWORK
                                                                                 571
                                                                                 572
C
      DO 20 J=1.ARCS
                                                                                 573
      Z=SUUNET(J.TSUUN)
                                                                                 574
      IF(5(Z).EQ.K) GO TO 21
                                                                                 575
      IF (T(Z).EQ.K) GO TO 21
                                                                                 576
20
      CONTINUE
                                                                                 577
C
                                                                                 578
C
         NCCE K IS NOT IN THIS SUBNETWORK
                                                                                 579
C
                                                                                 580
      GC TO 1
                                                                                 581
21
      CONTINUE
                                                                                 582
C
                                                                                 583
C
         NODE K IS IN THIS SUBNETWORK
                                                                                 584
C
                                                                                 585
      IF (K.EG.SOURCE) GO TO 1
                                                                                 586
      IF(K.EQ.SINK) GO TO 1
                                                                                 587
      NTARC=0
                                                                                 588
      APPCST=1
                                                                                 589
      DO 2 J=1.ARCS
                                                                                 590
      Z=SUENET(J.TSUBN)
                                                                                 591
      IF (S(Z).EQ.K) GO TO 2
                                                                                 592
      IF (T(Z).EQ.K) GO TO 2
                                                                                 593
      NTARC=NTARC+1
                                                                                 594
      TARC(NTARC)=SUBNET(J.TSUBN)
                                                                                 595
      CENTINUE
2
                                                                                 596
C
                                                                                 597
C
         TARC IS THE SUBNETWORK WITHOUT THE ARCS INVOLVING NODE K
                                                                                 598
C
         IF TARC CONTAINS A PATH FROM THE SOURCE TO THE SINK, THEN NODE
                                                                                 599
         K IS NOT A CUT NODE
C
                                                                                 600
c
         OTHERWISE. K IS A CUT NODE
                                                                                 601
C
                                                                                 602
      GRIGIN(1)=SUURCE
                                                                                 603
      PPCST(1)=SOURCE
                                                                                 604
      NCRIG=1
                                                                                 605
11
      CONTINUE
                                                                                 606
      NPOST=0
                                                                                 607
                                                                                 608
         IF THERE ARE NO ARCS IN THE TARC ARRAY. K IS A CUT NODE
C
                                                                                 609
```

```
C
                                                                                   610
      IF (NTARC.EQ.O) GO TC 44
                                                                                   611
C
                                                                                   612
c
         FIND ALL NODES WHICH COME AFTER AN ORIGIN: PUT THEM IN POST
                                                                                   613
c
         AND PPOST IF THEY ARE NOT PAST POSTS
                                                                                   614
c
                                                                                   615
      00 4 1=1 . NORIG
                                                                                   616
      Y=ORIGIN(1)
                                                                                  617
      DO 5 J=1.NTARC
                                                                                   618
      Z=TARC(J)
                                                                                   619
      U=5(Z)
                                                                                   620
      V=T(Z)
                                                                                   621
      IF (U.NE.Y) GO TO 5
                                                                                   622
C
                                                                                   623
         IF WE'VE REACHED THE SINK. NODE K IS NOT A CUT NODE
c
                                                                                   624
c
                                                                                   625
C
                                                                                   626
      IF (V.EQ.SINK) GO TO I
                                                                                   627
C
                                                                                   628
C
         IF V IS A PAST POST. LET'S IGNORE IT
                                                                                   629
C
                                                                                  630
      DG 49 L=1.NPPOST
                                                                                   631
      IF (V.EQ.PPOST(L)) GO TO 5
                                                                                   632
 49
      CONTINUE
                                                                                   633
      NFCST=NPOST+1
                                                                                   634
      PCST (NPGST)=V
                                                                                   635
      NPPOST=NPPOST+1
                                                                                   636
      PFOST (NPPOST) = V
                                                                                   637
      GC TC 5
                                                                                   638
      CONTINUE
                                                                                   639
5
4
      CCATINUE
                                                                                   640
C
                                                                                   641
c
          IF THERE ARE NOW NO POSTS. NODE K IS A CUT NODE
                                                                                   642
C
                                                                                   643
      IF (NPOST.NE.O) GO TO 13
                                                                                   644
44
      NCUT=NCUT+1
                                                                                   645
      RCLT(NCUT)=K
                                                                                   646
      GC TO 1
                                                                                   647
      NORIG=NPCST
                                                                                   648
13
      DO 14 L=1.NORIG
                                                                                   649
                                                                                  650
C
C
         THESE ARE NOW OUR NEW ORIGINS
                                                                                   651
                                                                                  652
C
      ORIGIN(L)=POST(L)
                                                                                   653
14
C
                                                                                  654
         CHECK THE NEW ORIGINS FOR THEIR POSTS
                                                                                   655
C
C
                                                                                   ú56
      GO TO 11
                                                                                   657
      CONTINUE
                                                                                   658
      IF (NCUT.EQ.0) GO TO 32
                                                                                   659
      ASUB=ASUB+1
                                                                                   660
      SCURC(NSUE)=SUURCE
                                                                                   661
      CRIGIN(1)=SOURCE
33
                                                                                   662
C
                                                                                   663
         NOW WE NEED TO FIND THE COMPONENTS OF THE SERIES SUBNETWORKS
C
                                                                                   664
C
         THAT ARE SEPARATED BY THE CUT NUDES
                                                                                   665
                                                                                   666
C
39
      SUMARC=0
                                                                                  667
      NORIG=1
                                                                                  668
      NEPOST=1
                                                                                  669
23
      NPCST=0
                                                                                  670
```

```
PPCST(1)=GRIGIN(1)
                                                                                   671
       DO 24 1=1.NURIG
                                                                                   672
       Y=CRIGIN(I)
                                                                                   673
       DO 25 J=1.ARCS
                                                                                   674
       Z=SUBNET(J.TSUBN)
                                                                                   675
 C
                                                                                   676
 c
          ALL ARCS BEGINNING AT THIS ORIGIN GO INTO THE NEW SUBNETWERK
                                                                                   677
 C
                                                                                   678
       IF (S(Z).NE.Y) GO TO 25
                                                                                   679
       SUMARC=SUMARC+1
                                                                                   680
       SUBNET (SUMARC. NSUB) = Z
                                                                                   681
 C
                                                                                   682
 c
          T(Z) WILL BE A POST IF IT ISN'T A REPEAT OF A CURRENT
                                                                                   683
 c
          ORIGIN OR A CUT NODE
                                                                                   684
 C
                                                                                   685
 c
          CHECK TO SEE IF IT IS A REPEAT
                                                                                   686
 c
                                                                                   687
                                                                                   688
       DO 30 K=1.NORIG
                                                                                   689
 C
                                                                                   690
 C
          IF T(Z) IS A REPEAT OF A CURRENT GRIGIN. LET'S IGNORE IT
                                                                                   691
 c
                                                                                   692
       IF(X.EG.ORIGIN(K)) GO TO 25
                                                                                   693
 30
       CONTINUE
                                                                                   694
 c
                                                                                   695
 C
          CHECK TO SEE IF T(Z) IS A CUT NODE. IF IT IS. DON'T PUT IT
                                                                                   696
 C
          IN THE POST OR PPOST ARRAYS
                                                                                   697
 C
                                                                                   698
       DO 52 W=1.NCUT
                                                                                   669
       D=RCUT(N)
                                                                                   700
       1F (X.NE.D) GJ TO 52
                                                                                   701
 C
                                                                                   702
          THIS CUT NODE IS THE SINK OF THE SUBNETWORK UNDER
 C
                                                                                   703
 C
          CONSIDERATION AND THE SOURCE OF THE NEXT SUBNETWORK TO BE
                                                                                   704
 c
          CONSIDERED.
                                                                                   705
 C
                                                                                   706
       SINKS(NSUB)=D
                                                                                   707
       SCURC(NSUB+1)=D
                                                                                   708
       GO TO 25
                                                                                   709
 52
       CCNTINUE
                                                                                   710
 C
                                                                                   711
 C
          IF T(Z) IS A REPEAT OF A PAST POST. LET'S IGNORE IT
                                                                                   712
 C
                                                                                   713
       DO 7 K=1. NPPOST
                                                                                   714
       IF (X.EG.PPOST(K)) GO TO 25
                                                                                   715
7
       CONTINUE
                                                                                   716
       NPOST=NPOST+1
                                                                                   717
       POST (NPOST) = X
                                                                                   718
       NPPOST=NPPOST+1
                                                                                   719
       PPOST(NPPOST)=X
                                                                                   720
 25
       CONTINUE
                                                                                   721
 24
       CCATINUE
                                                                                   722
 C
                                                                                   723
          IF WE HAVE NU POSTS LEFT. WE HAVE FOUND ALL OF THIS SUBNETWORK
 C
                                                                                   724
 C
                                                                                   725
       IF (NPOST-EQ.O) GO TO 34
                                                                                   726
       NORIG=NPOST
                                                                                   727
       00 28 L=1.NORIG
                                                                                   728
                                                                                   729
 28
       CRIGIN(L)=POST(L)
       GO TO 23
                                                                                   730
 34
       NAFCSS (NSUB) = SUMARC
                                                                                   731
```

```
ASUB=ASUB+1
                                                                                 732
      X=SGURC(NSUB)
                                                                                 733
c
                                                                                 734
C
         IF THE SOURCE OF NSUB IS NOT A CUT NODE. WE NEED TO ADJUST NSUB
                                                                                 735
c
         AND GO BACK TO THE MAIN PROGRAM FOR THE NEXT STAGE OF THE
                                                                                 736
C
         BREAKUP
                                                                                 737
C
                                                                                 738
      IF (X.EQ.0) GO TO 31
                                                                                 739
      ORIGIN(1)=X
                                                                                 740
      60 TO 39
                                                                                 741
31
      NSUB=NSUB-1
                                                                                 742
      SINKS(NSUB)=SINK
                                                                                 743
32
      TNSUB=NCUT+1
                                                                                 744
      RETURN
                                                                                 745
      END
                                                                                 746
      SUBROUTINE PUNSYN
                                                                                 747
C
         THIS SUBROUTINE PUNCHES THE CARDS NEEDED FOR SYNTHESIS
                                                                                 748
C
         CARD(1.1) = THE INSTRUCTION NUMBER.I
                                                                                 749
C
         CARD(1,2) = THE DECOMPGSABLE SUBNETWORK
                                                                                 750
c
         CARD(1,3) = ISORP;
                                                                                 751
c
              ISORP=0 IMPLIES A SERIES DECOMPOSITION
                                                                                 752
C
              ISORP=1 IMPLIES A PARALLEL DECOMPOSITION
                                                                                 753
         CARD(1.4...) = THE SUBNETWORKS RESULTING FROM THE
C
                                                                                 754
C
                         DECOMPOSITION OF SUBNETWORK I
                                                                                 755
      IMPLICIT INTEGER#2 (A-Z)
                                                                                 756
      COMMON /NEW/ CARD. INSTN
                                                                                 757
      INTEGER CARD(100.27). WHYNOT
                                                                                 758
      INTEGER
                                                 F5/13/
                                                                                 759
      SHYNOT = INSTN
                                                                                 760
      WEITE (F5) WHYNOT
                                                                                 761
      WRITE(F5) CARD
                                                                                 762
      ENCFILE F5
                                                                                 763
      RETURN
                                                                                 764
      END
                                                                                 765
      SUBROUTINE PRESUB(ISUBN)
                                                                                 766
                                                                                 767
C
         PREPARES THE SUBNETWORK FOR SUBNETWORK ANALYSIS
                                                                                 768
c
              INCLUDES RENUMBERING THE NODES 1.2.... CONSECUTIVELY.
                                                                                 769
C
                                                                                 770
      IMPLICIT INTEGER+2 (A-Z)
                                                                                 771
      COMMON SUBNET.S.T.SOURC.SINKS.NARCSS.NSUB.TNSUB
                                                                                 772
      INTEGER#2 S(1000).T(1000).SUBNET(1000.100).SCURC(100).SINKS(100).
                                                                                 773
     *NARCSS(100)
                                                                                 774
                                                                                 775
      INTEGER#2 HEAD(1000). TAIL(1000). NODE(1000)
      COMMGN/PREP/ NUMSUB
                                                                                 776
      INTEGER FILE3(1503).NET(3,500).N3
                                                                                 777
      INTEGER
                                  F3/11/
                                                                                 778
      EQUIVALENCE (FILE3(4), NET(1))
                                                                                 779
      NUMSUB=NUMSUB+1
                                                                                 780
                                                                                 781
c
         TSUBNETHE NUMBER OF THE MINIMUM SUBNETWORK
                                                                                 782
c
c
                                                                                 784
         RENUMBER THE NODES IN THE SUBNETWORK
C
                                                                                 785
                                                                                 786
      N=NARCSS (TSUBN)
                                                                                 787
      DC 1 1=1.N
                                                                                 788
      J=SUBNET([.TSUBN)
                                                                                 789
      NGDE (2+1-1) = S(J)
                                                                                 790
    1 NODE (2+1) = T(J)
                                                                                 791
C
                                                                                 792
```

```
793
C---- SORT THE NODES
                                                                               794
                                                                               795
  100 LIMIT = 2*N
      1PASS = 2*N-1
                                                                               796
                                                                               797
      IFLG = 1
                                                                               798
      CC 2 J=1. IPASS
      LIMIT = LIMIT-1
                                                                               799
      IF(IFLG.EG.O) GO TO 3
                                                                               800
      IFLG = 0
                                                                               801
                                                                               802
     DC 4 K=1.LIMIT
      IF (NUDE(K).LE.NODE(K+1)) GO TO 4
                                                                               803
                                                                               804
      KEEP = NODE (K+1)
                                                                               805
      NODE (K+1) = NODE (K)
                                                                               806
      NCDE (K) = KEEP
     IFLG = 1
                                                                               807
    4 CONTINUE
                                                                               808
                                                                               809
    2 CONTINUE
                                                                               810
C---- DISCARD ANY DUPLICATIONS IN THE SORTED LIST
                                                                               811
C
                                                                               812
    3 KOUNT=1
                                                                               813
     DO 5 I=1.IPASS
                                                                               814
      IF (NGDE(I).EQ.NODE(I+I)) GO TO 5
                                                                               815
      KOUNT = KOUNT+1
                                                                               816
     NGDE(KOUNT) = NODE(I+1)
                                                                               817
    5 CENTINUE
                                                                               818
C
                                                                               819
C----CHANGE INPUT-HEAD. TAIL
                                                                               820
                                                                               821
C
      DC 8 L=1.N
                                                                               822
      I = SUBNET (L. TSUBN)
                                                                               823
                                                                               824
      DO 9 J=1.KOUNT
      IF ( T(1).NE.NODE(J)) GC TO 9
                                                                               825
                                                                               826
      HEAD(L)=J
      GO TO 8
                                                                               827
    9 CONTINUE
                                                                               828
    8 CONTINUE
                                                                               829
      DC 11 L=1.N
                                                                               830
      I=SUBNET(L.TSUBN)
                                                                               831
      DC 12 J=1.K
                                                                               832
      IF ( S(1).NE.NODE(J)) GO TO 12
                                                                               833
      TAIL(L)=J
                                                                               834
      GC TO 11
                                                                               835
  12 CONTINUE
                                                                               836
   11 CENTINUE
                                                                               837
      FILE3(1) = N
                                                                               838
      FILE3(2) = TSUBN
      FILE3(3) = KOUNT
                                                                               840
      DO 10 J=1.N
                                                                               841
      NET(1.J) = HEAD(J)
                                                                               842
      NET(2.J) = TAIL(J)
                                                                               843
10
      NET(3.J) = SUBNET(J.TSUBN)
                                                                                844
                                                                               845
C
C
         OUTPUT
                                                                               846
                                                                               847
C
      WRITE(F3) FILE3(1)
                                                                               848
      N3 = 2 + 3+N
                                                                               849
      CALL WRT (F3.FILE3(2).N3)
                                                                               850
      WRITE(6.200)
                                                                               851
      FORMAT (1HO./16X. THE FOLLOWING REPRESENTATION OF THE SUBNETWORK AB
                                                                               852
200
     OVE WILL BE USED BY THE SUBNETWORK ANALYSIS PROGRAM: 1)
                                                                               853
```

6

	WRITE(6,30)	854
30	FORMAT(1H0.19X. ACTIVITY .4X. TAIL .6X. HEAD)	85
	WRITE(6.31) (1.TAIL(1).HEAD(1).I=1.N)	85
31	FORMAT(20x.15.5x.15.5x.15)	85
	WRITE(6.7005)	85
7005	FORMAT(1H1)	859
	RETURN	86
	END	861
	SUBROLTINE BRT(J.A.N)	863
	DIMENSION A(N)	863
	brite(J) A	864
	RETURN	869
	END	866
	SUBROUTINE RD(J.A.N)	86
	DIMENSION A(N)	869
	READ(J) A	869
	RETURN	870
	END	87

7. SUBNET

```
C
         SUBNETWORK ANALYSIS PROGRAM
C
                                                                             2
         THIS PROGRAM DETERMINES THE DURATION DISTRIBUTION FOR EACH
                                                                              3
c
C
        SUBNETWORK IDENTIFIED BY THE DECOMPOSITION PROGRAM.
C
C
        FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET
C
        MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL
                    PROJECT NETWORK
C
                                                                             8
        LMAX = THE MAXIMUM NUMBER OF ACTIVITIES ON A CRITICAL PATH
C
c
                    LMAX IS LESS THAN OR EQUAL TO MSMAX
                                                                             10
C
        CMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN A CLUSTER
                                                                             11
                    CMAX IS LESS THAN OR EQUAL TO MSMAX
C
                                                                             12
C
        MSMAX = THE MAXIMUM NUMBER OF ACTIVITIES ALLOWED IN ANY ONE
                                                                             13
C
                    SUBNETWORK
                                                                             14
        NSUBMAX = THE MAXIMUM NUMBER OF SUBNETWORKS ALLOWED IN THE
C
                                                                             15
C
                    DECOMPOSITION PROCESS
                                                                             16
c
        IEDFMAX = THE MAXIMUM NUMBER OF SUBDIVISIONS ALLOWED IN THE
                                                                             17
C
                    APPROXIMATE COF FOR EACH SUBNETWORK
                                                                             18
        SCIM = 12 * (IEDFMAX-1) + 2
                                                                             19
C
        CURRENTLY. MMAX=1000; LMAX=50; CMAX=100; MSMAX=500; NSUBMAX=100
C
                                                                             20
C
                    IEDFMAX=20; SDIM=230
                                                                             21
C
                                                                             22
C
     DIMENSION CTIME(MSMAX).DTIME(MSMAX).ADRD(MSMAX).TAIL(MSMAX).
                                                                             23
               HEAD(MSMAX).XNODE(MSMAX).IBB(LMAX).KB(LMAX).ICRITP(LMAX)
C
                                                                             24
C
               COT(MSMAX).SIGMA(MSMAX).NINCL(LMAX).INCLUS(LMAX.CMAX).
                                                                             25
               FD(IEDFMAX). ULEFD(IEDFMAX). LLEFD(IEDFMAX). FHI(MSMAX).
C
                                                                             26
               FLU(MSMAX), PP(MSMAX), PQ(MSMAX), FLHAT(IEDFMAX).
                                                                             27
C
               FUHAT (IEDFMAX). FDLHAT (3, IEDFMAX). FDUHAT (3, IEDFMAX).
                                                                             28
C
               FHAT(IEDFMAX). BRHS1(SDIM). INBASE(SDIM). XB1(SDIM).
                                                                             29
c
               Y1(SDIM).81INV(SDIM.SDIM).FILE3(3+MSMAX+2).NET(3.MSMAX).
                                                                             30
               LUP(3.MMAX)
C
                                                                             31
                                                                             32
        IN SUBROUTINE ORDER THE ARRAYS ARE:
C
                                                                             33
C
     DIMENSION ND(MSMAX) . NDD(MSMAX)
                                                                             34
C
                                                                             35
C
        IN SUBROUTINE CLUSTR THE ARRAYS ARE:
                                                                             36
C
     DIMENSION LEFT(MSMAX).LEFTO(MSMAX).NONCP(MSMAX).NINAG(MSMAX).
                                                                             37
               NCLINC(LMAX). ASSGRP(LMAX, LMAX). CLINCL(LMAX, LMAX),
C
                                                                             38
C
               EGRP(LMAX)
                                                                             39
C
                                                                             40
41
      IMPLICIT REAL+8 (A-H.O-Z)
                                                                             42
     COMMON /BLKA/CTIME.DTIME.M
                                                                             43
     DIMENSION CTIME(0500), DTIME(0500)
                                                                             45
     COMMON /BLAA/XNODE.TAIL.HEAD.AORD.NMM
                                                                             46
      INTEGER ACRD(500), TAIL(500), HEAD(500)
                                                                             47
     DIMENSION XNODE(500)
                                                                             48
. . . . . . . . . . . . . . . .
                                                                             49
     CCMMCN /BLKB/IBB.KB.KKB.KKK
                                                                             50
C
     DIMENSION 188(50).KB(50)
                                                                             52
     COMMON /BLKC/ICRITP.KCPB
                                                                             53
     DIMENSION (CRITP(50)
                                                                             54
                                                                             55
CCMMCN /BLKD/SIGMA.CUT
                                                                             56
     DIMENSION COT(500).SIGMA(500)
                                                                             57
                                                                             58
COMMON /BLKE/INCLUS.NINCL.NCLUS
                                                                             59
     DIMENSION NINCL(50) . INCLUS(50.100)
```

2	•	•	COMMON /BLKG/FD.ULEFD.LLEFD REAL+8 FC(20).ULEFD(20).LLEFD(20)	61 62 63
-				64
•	•	•	COMMON /BLKH/FHI.FLO.PP.PQ	65
			DIMENSION FHI(500).FLO(500).PP(500).PQ(500)	66
C			••••••	67
			COMMON /BLKI/IEDF.NSUB.SAMSIZ	68
			INTEGER SAMSIZ	69
0	•			70
			CCMMCN /BLKJ/THET.LAMBD	71
			REAL+8 LAMBO	72
C		•		73
			COMMON /BLKK/IR.IB	74
C	•	•		75
			COMMON /BLKL/FLHAT.FUHAT	76
			DIMENSIGN FLHAT(20), FUHAT(20)	77
C	•	•		78
			COMMON /BLKS/SUM. ICLGIB	79
	•	•		80
			CCMMON /BLKY/FDUHAT.FDLHAT.FHAT	81
_			DIMENSION FOLHAT (3, 20), FOUHAT (3, 20), FHAT (20)	82
-	•	•	COMMON /BLKZ/THETA	84
			REAL+4 THETA(3)	85
-				86
-	•	•	COMMON /BLK1/B1INV.OMEGA.MM.KK.MMKK.MMKK2.N.MP1.JFIRST	87
-			CCMMON /BLK2/Y1	88
-			DIMENSION BRHS1(230). INBASE(230).XB1(230).Y1(230).B1[NV(230.230)	89
				90
		•	INTEGER THELAM	91
			INTEGER FILE3(1502)	92
			INTEGER F3/11/.F4/12/.F5/13/	93
			REAL+8 LUP(3,1000)	94
			DIMENSION NET(3.500)	95
			DIMENSION NIMPT(3)	96
			DIMENSIGN CDF(12)	97
			REAL+4 LAMBDA(3)	98
			REAL+4 RTHETA(3).RLAMBD(3)	99
			REAL+8 M1.M2.M3	100
			EQUIVALENCE (FILE3(3).NET(1))	101
			EGUIVALENCE (FILE3(6).RTHETA(1)).(FILE3(9).RLAMBD(1))	102
	•	•		103
-			LOTTE	104
			WRITE(6,90001)	105
90	000		FCRMAT(1H1.132(***)/1X.132(***)/*OTHIS IS THE OUTPUT FROM THE SUBN	106
			FETWORK ANALYSIS PROGRAM: SUBNET*/1H0.132(***)/1x.132(***)//// FOR EACH SUBNETWORK*.1H*.*S APPROXIMATE DURATION DISTRIBUTION IS DET	108
			FERMINED.)	109
				110
2		**		111
c			READ IN INFORMATION PERTAINING TO ALL SUBNETWORKS	112
				113
				114
			REWIND F3	115
			REWING F5	116
			READ(F5)	117
			READ(F5)	118
			REAC(F5)	119
			REWIND F4	120
			READ(F4)	121

```
READIFAL NACT
                                                                   122
                                                                   123
C
                                                                   124
       DETERMINE L.U. AND P FOR EACH ACTIVITY IN THE SIMPLIFIED
c
                                                                   125
C
       PROJECT NETWORK
C
                                                                   127
128
     DO 10 L=1.NACT
                                                                   129
     READ (F4)CDF
                                                                   130
     M1=0.D0
                                                                   131
     M2=0.D0
                                                                   132
     M3=0.00
                                                                   133
     DO 20 K=2.11
                                                                   134
     MI=MI+COF(K)
                                                                   135
     M2=M2+CDF(K) ++2
                                                                   136
     M3=M3+CDF(K)++3
20
                                                                   137
     M1=M1+.100
                                                                   138
     M2=M2+.1D0
                                                                   139
     M3=M3+.100
                                                                   140
     T3=M1+M1-M2
                                                                   141
     IF (DABS (T3) .GE . 1 .D-101GO TO 21
                                                                   142
     U=CDF(2)
                                                                   143
     B=U
                                                                   144
     P=1.00
                                                                   145
     GO TO 22
                                                                   146
21
     T1=(M3-M1+M2)++2
                                                                   147
     T2=4.D0+T3+(M2+M2-M1+M3)
                                                                   148
     T3=2.00+T3
                                                                   149
     U=-#3+#1 ##2-DSQRT(T1-T2)
                                                                   150
     U=U/13
                                                                   151
     B= (M2-M1+U)/(M1-U)
                                                                   152
     P=(M1-U)/(B-U)
                                                                   153
     LUP(1.L)=B
22
                                                                   154
     LUF(2.L)=U
                                                                   155
     LUP(3.L)=P
                                                                   156
10
     CONTINUE
                                                                   157
     CALL RD(F3.FILE3.11)
                                                                   158
     IEDF=FILE3(1)
                                                                   159
     NMAX=FILE3(2)
                                                                   160
     IPCCL=FILE3(3)
                                                                   161
     SAMSIZ=FILE3(4)
                                                                   162
     THELAM=FILE3(5)
                                                                   163
     00 2 1=1.3
                                                                   164
     THETA(1)=RTHETA(1)
                                                                   165
     LAMBDA(I)=RLAMBD(I)
                                                                   166
     CONTINUE
                                                                   167
168
C
                                                                   169
       ANALYZE EACH SUBNETWORK SEPARATELY
C
                                                                   170
C
                                                                   171
C
                                                                   172
c
       REAC IN THE DESCRIPTION OF THE SUBNETWORK
                                                                   173
C
                                                                   174
                                                                   175
READ(F3.END=999) M
                                                                   176
     N2 = 2 + 3+M
                                                                   177
     CALL RD(F3.FILE3(1).N3)
                                                                   178
     NSUB = FILE3(1)
                                                                   179
     APP = FILE3(2)
                                                                   180
     WEITE (6.2600) NSUB
                                                                   181
2600 FORMAT(1H1. 5x. SUBNETWORK .. 15)
                                                                   182
```

```
M = THE NUMBER OF ACTIVITIES IN THE SUBNETWORK NMM = THE NUMBER OF NODES IN THE SUBNETWORK
                                                                           183
                                                                           184
      MP1=M+1
                                                                           185
C
                                                                           186
        THE ACTIVITIES ARE DESCRIBED IN TERMS OF THEIR NODES
                                                                           187
C
        II=THE TAIL NODE. THE ORIGIN NODE
                                                                           188
        JJ=THE HEAD NODE. THE TERMINAL NODE
                                                                           189
C
        FLC = THE LOWER POINT
                                                                           190
C
        FHI = THE UPPER POINT
                                                                           191
C
        SIGMA = (FHI - FLO) +DSQRT( P+(1-P) ) = STD. DEVIATION
                                                                           192
C
        PP = THE PROBABILITY OF THE LOWER POINT
                                                                           193
c
                                                                           194
     DO 610 I=1.M
                                                                           195
     HEAD(I) = NET(1.I)
                                                                           196
      TAIL(1) = NET(2.1)
                                                                           197
             = NET(3.1)
                                                                           198
     FLC(1) = LUP(1.K)
                                                                           199
     FHI(I) = LUP(2.K)
                                                                           200
             = LUP(3.K)
     PP(I)
                                                                           201
     PQ(1)=1.00-PP(1)
                                                                           202
     SIGMA(1)=(FHI(1)-FLO(1))+DSQRT( PP(1)+PQ(1))
                                                                           203
     CGT(1)=PP(1)*FLO(1)*PG(1)*FHI(1)
                                                                           204
     CTIME(I) = COT(I)
                                                                           205
610
     CONTINUE
                                                                           206
C
        COT = THE ORIGINAL RIGHT-HAND SIDES. I.E. THE MEANS
                                                                           207
C
              OF FLO AND FHI
                                                                           208
C
        CTIME = THE CURRENT RIGHT-HAND SIDES
                                                                           209
     WRITE(6.2700)
                                                                           210
     FORMAT(1HO.10X. INITIAL INPUT')
                                                                           211
      WRITE(6.2701)
                                                                           212
2701 FORMAT(1HO.IDX. ACTIVITY ORIGIN TERMINAL LOWER POINT UPPER P
                                                                           213
     *CINT
              MEAN
                     STANDARD DEVIATION
                                         PROB. LOWER PT. ..
                                                                           214
     DC 2704 I=1.M
                                                                           215
     WRITE(6.2702) 1.TAIL(1).HEAD(1).FLO(1).FHI(1).CTIME(1).SIGMA(1).PP
                                                                           216
    *(1)
                                                                           217
2702 FORMAT(1H .13X.13.5X.13.7X.13.5X.F10.4.3X.F10.4.3X.F10.4.4X.F10.4.
                                                                           218
    #14X,F7.5)
                                                                           219
220
C
                                                                           221
        IF THERE IS ONLY 1 ACTIVITY IN THE SUBNETWORK. LET FHAT BE
C
                                                                           222
C
        THAT ACTIVITY'S DURATION DISTRIBUTION
                                                                           223
                                                                           224
225
      IF (M.GT.1)GO TO 4
                                                                           226
     RENIND F4
                                                                           227
      1=K+1
                                                                           228
     DO 25 L=1.1
                                                                           229
     READ(F4)
                                                                           230
25
     CONTINUE
                                                                           231
     READ(F4)( FD(1).1=1.12)
                                                                           232
      JEDF=12
                                                                           233
     FHAT(1)=0.00
                                                                           234
      FHAT (12)=1.00
                                                                           235
     M1=-.05D0
                                                                           236
     DO 30 L=2.11
                                                                           237
      M1=M1+-1D0
                                                                           238
     FHAT(L)=M1
                                                                           239
30
      WRITE(6.700)NSUB
                                                                           240
700
     FCRMAT(IH1.5X. THE DURATION DISTRIBUTION FOR SUBNETWORK .. 12)
                                                                           241
      WRITE(6.108)
                                                                           242
      DG 701 I=1.JEDF
                                                                           243
```

```
701
    WRITE(6.103)FD(1).FHAT(1)
                                                          244
    GO TO 3
                                                          245
246
c
                                                          247
      INITIAL DETERMINES THE CRITICAL PATH WHEN EACH
C
                                                          248
      ACTIVITY'S DURATION IS ITS MEAN. LOWER BOUND. AND UPPER BOUND
c
                                                          249
C
                                                          250
251
   CALL INITAL
                                                          252
4
253
c
                                                          254
      IF M IS LESS THAN OR EQUAL TO NMAX. LET FHAT BE THAT
C
                                                          255
C
      ACTIVITY'S DISCRETE DURATION DISTRIBUTION
                                                          256
C
                                                          257
258
    IF (M.GT.NMAX)GO TO 5
                                                          259
    CALL FINDE
                                                          260
    JEDF= IEDF
                                                          261
    GC TC 3
                                                          262
263
c
                                                          264
C
      DETERMINE THETA AND LAMBDA
                                                          265
C
                                                          266
      IF THELAM = 0 THE DEFAULT PAIRS (THETA.LAMBDA)=(1.1).(2.2).(3.2
                                                          267
C
c
      WILL BE USED
                                                          268
C
                                                          269
      IF THELAM IS NOT O THE USER-SPECIFIED (THETA. LAMBDA) PAIRS
C
                                                          270
c
      WILL BE USED
                                                          271
C
                                                          272
      NOTE: THE FOLLOWING RELATIONS MUST ALWAYS BE TRUE
C
                                                          273
C
          THETA(1)<=THETA(2)<=THETA(3)
                                                          274
C
          LAMBDA(1) <= LAMBDA(2) <= LAMBDA(3)
                                                          275
C
                                                          276
277
    DO 500 I=1.3
                                                          278
    IF (THELAM.NE.OJGO TO 222
                                                          279
    LAMBD=2.DO
                                                          280
    IF (1.EQ.1)LAMBD=1.00
                                                          281
    THET=DFLOAT(1)
                                                          282
    THETA(1)=THET
                                                          283
    GC TO 201
                                                          284
    THET=THETA(1)
222
                                                          285
    LAMBD=LAMBDA(I)
                                                          286
                                                          287
C
                                                          288
C
      SUBROUTINE CLUSTR DETERMINES THE CLUSTER FOR GIVEN THETA AND
                                                          289
C
      LAMBDA
                                                          290
                                                          291
C
292
   CALL CLUSTR
                                                          203
201
C
                                                          294
      THE FOLLOWING DETERMINES WHETHER CR NOT THESE CLUSTERS HAVE
C
                                                          295
C
      BEEN CONSIDERED BEFORE
                                                          296
C
                                                          297
    I SUM= 0
                                                          298
    DO 200 11=1.NCLUS
                                                          299
    ISUM= ISUM+NINCL(II)
200
                                                          300
    NIMPT(I)=ISUM
                                                          301
                                                          302
    IF(1.EQ.1)GO TO 202
    IF(ISUM.EQ.NIMPT(I-1))GO TO 203
                                                          303
202
    CONTINUE
                                                          304
```

```
306
       CALCULATE THE UPPER AND LOWER BOUNDS ON THE SUBNETWORK
C
                                                                    307
     CURATION DISTRIBUTION
                                                                    308
                                                                   309
310
C
       ELIMINATE FROM THE CLUSTERS ANY ACTIVITIES HAVING CONSTANT
C
                                                                   312
C
       CURATION. IF THESE ACTIVITIES ARE LEFT IN. THE TOTAL NUMBER
                                                                   313
      OF POSSIBLE ACTIVITY DURATION CONFIGURATIONS FOR THE CLUSTERS
C
                                                                   314
C
       IS ARTIFICIALLY INFLATED.
                                                                   315
C
                                                                   316
     DO 250 11=1.KCPB
                                                                   317
     IL=NINCL(II)
     IF(IL.LE.1)GO TO 250
                                                                   319
     1 1 1 = 1
                                                                   320
251
     IF(III.GT.ILIGO TO 254
     L=INCLUS(II.III)
                                                                   322
     IF(SIGMA(L).NE.O.DOJGO TO 253
                                                                   323
     IL=IL-1
                                                                   324
     IF ( 111. GT . IL ) GO TO 254
                                                                   325
     DC 252 1111=111.1L
                                                                   326
252
     INCLUS(11.1111)=INCLUS(11.1111+1)
                                                                   327
253
     111111111
                                                                   328
     GC TG 251
                                                                   329
     NINCL(II)=IL
254
     IF (IL.EQ.O) NCLUS=NCLUS-1
                                                                   331
     CONTINUE
                                                                   332
     IF(IPOOL.EQ.1) CALL UNION
                                                                   333
     IF (IPOOL.EQ.O) CALL SINGLE
                                                                   334
C
                                                                   336
C
       SAVE THE ESTIMATES CALCULATED BY UNION OR SINGLE
                                                                   337
C
       FOR THE 1-TH THETA-LAMBOA PAIR
                                                                   338
C
                                                                   339
203 CONTINUE
                                                                   341
     WRITE(6.100) NSUB.THET.LAMBD
                                                                   342
     FORMAT(1H1. 5x. BOUNDS ON THE DURATION DISTRIBUTION FOR SUBNETWORK
100
                                                                   343
          14." WHEN (THETA, LAMBDA) = ( ".E15.5." . ".E15.5." )")
                                                                   344
     BRITE(6.101)
                                                                   345
     FGRMAT(1HO.14X. TIME .16X. LOWER BOUND .9X. UPPER BOUND ./)
101
                                                                   346
     IFLG=0
                                                                   347
     DC 300 II=1.IEDF
                                                                   348
     FLTEMP=FLHAT(II)
                                                                   349
     FUTEMP=FUHAT(II)
                                                                   350
     WRITE(6.103)FD(11).FLTEMP.FUTEMP
                                                                   351
103
     FCRMAT(1H .10X.3(E15.5.5X))
                                                                   352
     IF (FL TEMP . NE . FUTEMP) IFLG=1
                                                                   353
     FHAT(II)=FLTEMP
                                                                   354
     FOLHAT (I.II)=FLTEMP
                                                                   355
     FCUHAT( I . II ) = FUTEMP
     IF(IPOOL.EQ.1) WRITE(6.350)
                                                                   357
     IF ( IPOOL . EQ . 0 ) WRITE ( 6.351)
                                                                   358
     FCFMAT(1HO./11x. THE BOUNDS WERE DETERMINED USING THE UNION-CLUSTE
350
                                                                   359
    *R PROCEDURE. 1)
                                                                   360
     FGFMAT(1HO./11X. THE BOUNDS WERE DETERMINED USING THE MAXIMUM-CLUS
                                                                   361
    TER PROCEDURE. 1)
                                                                   362
     IF(ICLG18)352.353.353
                                                                   363
352
     WRITE (6.6645)
                                                                   364
6645 FORMAT(1HO./11X. ALL ACTIVITY DURATION CONFIGURATIONS WERE CONSIDE
                                                                   365
```

```
*REC. 1
                                                                           366
     GO TO 500
                                                                           367
353
      IF (IPOOL)357.357.358
                                                                           368
                                                                           369
350
     WEITELE. 6C461 SAMSIZ
6046 FURMAT(1HO./11X. ONLY ... IB. OF THE POSSIBLE ACTIVITY DURATION CON
                                                                           370
    *FIGURATIONS WERE CONSIDERED. 1)
                                                                           371
      IF( | CLG | B.EQ. 0) WRITE (6.105)
                                                                           372
                                                                           373
     IF (ICLGIB.EQ.1) WRITE(6.106)
     GO TO 500
                                                                           374
357
     WRITE(6,6047)SAMSIZ
                                                                           375
6047 FCRMAI(1H0./11X. ONLY .. 18. CF THE POSSIBLE ACTIVITY DURATION
                                                                           376
     *CONFIGURATIONS FOR EACH CLUSTER WERE CONSIDERED. *)
                                                                           377
                                                                           378
     IF(ICLGIB.EG.O) WRITE(6.105)
     IF (ICLGIB.EQ.1) WRITE(6,106)
                                                                           379
     FORMAT(1HO.10X. SYSTEMATIC SAMPLING WAS USED. )
                                                                           380
105
106
     FORMAT(1HO.10X. SIMPLE RANDOM SAMPLING WAS USED. )
                                                                           381
500
     CCNTINUE
                                                                           382
                                                                           383
C
        IF THE BEST UPPER AND LOWER BOUNDS COMPUTED ABOVE ARE EQUAL.
                                                                           384
        THERE IS NO NEED TO PERFORM THE EXTRAPOLATION
                                                                           385
C
C
                                                                           386
C
        IFLG = 0 IMPLIES THE BOUNDS ARE EQUAL
                                                                           387
        IFLG = 1 IMPLIES THE BOUNDS ARE NOT EQUAL
                                                                           388
C
C
                                                                           389
     IF (IFLG.EG.1)GO TO 802
                                                                           390
801
                                                                           391
     WRITE(6.803)NSUB
     FORMAT(1H1.5x. THE UPPER AND LOWER BOUNDS ON THE DISCRETE SUBNETNO
803
                                                                           102
    *RK DURATION DISTRIBUTION ARE EQUAL. * //6x, * THUS AN ESTIMATE OF THE
    *DISCRETE DURATION DISTRIBUTION FOR SUBNETWORK *, 14, * IS*)
                                                                           394
     JECF = LEDF
                                                                           395
     GC TC 804
                                                                           396
397
C
                                                                           398
C
        SLEROUTINE SIMPLX INTERPOLATES BETWEEN THE UPPER AND LOWER
                                                                           366
C
        BOUNDS DETERMINED ABOVE AND COMPUTES AN ESTIMATED COMPLETION
                                                                           400
C
        TIME DISTRIBUTION FOR THE SUBNETWORK BEING CONSIDERED
                                                                           401
                                                                           402
C
403
802
     CALL SIMPLX
                                                                           404
     JECF= LEDF
                                                                           405
      BRITE (6.107) NSUB
                                                                           406
     FORMAT(1H1.5X. 'AN ESTIMATE OF THE DISCRETE DURATION DISTRIBUTION F
107
                                                                           407
    OR SUENETWORK .. 121
                                                                           408
804
                                                                           409
     WRITE(6.108)
     FORMAT (1HO.14X. TIME . 16X. DISTRIBUTION /)
108
                                                                           410
     DO 9471 I=1.JEDF
                                                                           411
9471
     WRITE(6.103)FD(1).FHAT(1)
                                                                           412
                                                                           413
3
     WRITE(F5)NSUB.JEDF.(FD(1).FHAT(1).[=1.JEDF)
                                                                           414
     GC TC 1
                                                                           415
999
     WRITE (6.850)
                                                                           416
850
     FCFMAT(1H1)
                                                                           417
      ENOFILE F5
                                                                           418
     STOP
                                                                           419
     END
                                                                           420
     SCEROUTINE RD(J.A.N)
                                                                           421
     DIMENSION A(N)
                                                                           422
     READ(J) A
                                                                           423
                                                                           424
     RETURN
     END
                                                                           425
     SCERCUTINE INITAL
                                                                           426
```

	The second secon
	INITIAL DETERMINES THE CRITICAL PATH WHEN EACH
c	ACTIVITY'S DURATION IS ITS MEAN, LOWER BOUND, AND UPPER BOUND
c	
· · ·	IMPLICIT REAL+8 (A-H,O-Z)
	COMMON /BLKA/CTIME.DTIME.M
	DIMENSION CTIME(0500), DTIME(0500)
	COMMON /BLKB/IBB.KB.KKB.KKK
	CIMENSION (88(50) KB(50)
	CCMMCN /BLKC/ICRITP.KCPB
	DIMENSION ICRITP(50)
	CCMMCN /BLKG/FD.ULEFD.LLEFD
	REAL+8 FD(20).ULEFD(20).LLEFD(20)
	CCMMON /BLKH/FHI .FLO.PP.PG
	DIMENSION FHI (500) .FLO(500) .PP(500) .PQ(500)
	COMMON ADMINISTRATION NAMED CAMEET
	COMMON /BLKI/IEDF.NSUB.SAMSIZ INTEGER SAMSIZ
	CALL GROER
	TOTAL=CPTIME(CPATHT)
	CALL FPATH
C	ICRITP(L)= THE L-TH ACTIVITY ON THE CRITICAL PATH
C .	KCPB= THE NUMBER OF ACTIVITIES ON THE CRITICAL PATH
•	KCPB=KKB
	OC 2802 I=1.KCPB
2802	ICRITP(1)=188(1)
	X=TCTAL .
	WRITE(6.851) X
851	FORMAT(1HO.5X. THE CRITICAL PATH TIME WHEN EACH ACTIVITY'S COMPLE
	TION TIME IS SET EQUAL TO ITS MEAN IS = ".DI5.5)
76.06	FORMAT(1H0.10X. THE .13. NGDES ON THE CRITICAL PATH ARE AS FOLLO
	*NS BEGINNING WITH THE TERMINAL NODE: 1)
	WRITE(6.7707) (KB(I).I=1.KKK)
7707	FCRMAT(15X-20(13-'-'))
	WRITE(6.7710) KKB
	FORMAT(1HO.10X, THE .13, CRITICAL ACTIVITIES ARE AS FOLLOWS BEGI
	NAING WITH THE TERMINAL ACTIVITY:) WRITE(6.7707) (IBB(I).I=1.KKB)
c	WHITE(O, //U/) (IDD(I/) I=I,KKD)
č	THE COMPLETION TIME FOR ALL ACTIVITIES IS SET TO THEIR LOWER
c	PERCENTILE. THE RESULTING LONGEST PATH TIME IS A LOWER
c	BOUND ON THE SUBNETWORK DURATION TIME.
c	
	DC 6402 I=1.M
6402	CTIME(1) = FLO(1)
	CPLB=CPTIME(CPATHT)
	WRITE(6.6405) CPLB
6405	FORMAT(1H0.5X. "A LOWER BOUND ON THE SUBNETWORK DURATION TIME IS =
	•'.E15.5)
	BRITE(6,8606) KKK
8606	FORMAT(1h0.10x. THE 1.13. NODES ON THE LONGEST PATH ARE AS FOLLOW

```
*S BEGINNING WITH THE TERMINAL NODE: ")
                                                                       488
     WRITE(6.7707) (KB(1).1=1.KKK)
                                                                       489
     WRITE(6.8710) KKB
                                                                       450
8710 FORMAT(ING.IOX. THE '. 13. ACTIVITIES UN THE LUNGEST PATH ARE AS F
                                                                        491
    OLLOWS BEGINNING WITH THE TERMINAL ACTIVITY: .)
                                                                       492
     WRITE(6.7707) (188(1).1=1.KKB)
                                                                        493
C
                                                                       494
c
        THE COMPLETION TIME FOR ALL ACTIVITIES IS SET TO THEIR UPPER
                                                                        495
c
        PERCENTILE. THE RESULTING LONGEST PATH TIME IS AN UPPER BOUND
                                                                       496
C
        CN THE SUBNETWORK DURATION TIME.
                                                                       497
C
                                                                        498
     DO 6406 I=1.M
                                                                       400
6406 CTIME(1) = FHI(1)
                                                                       500
     CPUB=CPTIME(CPATHT)
                                                                       501
     CALL FPATH
                                                                       502
     WRITE(6.6409) CPUB
                                                                        503
6409 FORMAT(1H0.5x. A UPPER BOUND ON THE SUBNETWORK DURATION TIME IS =
                                                                       504
    ** .E15.5)
                                                                       505
     WRITE(6.8606) KKK
                                                                       506
     #RITE(6.7707) (KB(1).1=1.KKK)
                                                                       507
     WAITE (6.8710) KKB
                                                                       508
     WRITE(6.7707) (188(1).1=1.KKB)
                                                                       509
C
C
        FD(1) = THE LOWER BOUND ON THE SUBNETWORK DURATION TIME
                                                                       511
C
               PLUS THE 1-TH FRACTION OF THE DISTANCE TO THE UPPER
                                                                       512
C
               BOUND
                                                                       513
C
                                                                       514
C
        FO IS USED TO BUILD AN 'EMPIRICAL' DISTRIBUTION OF THE
                                                                       515
C
        SUBNETWORK DURATION TIMES
                                                                       516
c
                                                                       517
     C=CPUH-CPLB
                                                                       518
     CC=.8C0*C
                                                                       519
     LEDF = . 6DO + IEDF
                                                                       520
     CC=CC/LEDF
                                                                       521
     DC 6412 I=1.LEDF
6412 FD(1)=CPLB+1+CD
                                                                       523
     CD=(C-CC)/(IEDF-LEDF)
                                                                       524
     11=LEOF+1
                                                                       525
     C=CPLB+CC
                                                                       526
     OC 6512 1=11.1EDF
                                                                       527
6512 FD(1)=C+(1-LEDF)+CD
                                                                       52A
     RETURN
                                                                       529
     END
                                                                       530
     SUBROUTINE ORDER
                                                                       531
        THIS SUBROUTINE DETERMINES THE ORDER IN WHICH TO CONSIDER
C
                                                                       532
        THE ACTIVITIES FOR THE CALCULATION OF THE LONGEST PATH TIME
                                                                       533
       534
     IMPLICIT REAL+8 (A-H.O-Z)
                                                                       535
     COMMON /BLKA/CTIME.DTIME.M
                                                                       536
     DIMENSION CTIME(0500).DTIME(0500)
                                                                       537
                                                                       538
     COMMON /BLAA/XNODE.TAIL.HEAD.ACRD.NMM
                                                                       539
     INTEGER ADRD(500). TAIL(500). HEAD(500)
                                                                       540
     DIMENSION XNUDE(500)
                                                                       541
542
     DIMENSION ND(500).NDD(500)
                                                                       543
544
                                                                       545
     NDD(1)=1
                                                                       546
     DO 5 1=2.N
                                                                       547
     NCC(1)=0
                                                                       548
```

	00 6 1=1.M	549
6	ACRD(I)=0 K=0	550
		551
	MP=M+1	552
	DO 1 II=1,MP	553
20	DC 20 1=1.N NO(1)=NDD(1)	554
20		555
	III=0	556
	IP=II+1	557
	DO 2 J=1.M	558
	IF(NC(TAIL(J)).NE.II) GO TO 2	559
	NCD(HEAD(J))=IP	560
	III=1	561
	IF(K.EQ.0) GO TO 14	562
	00 10 L=1.K	563
	IF(AORD(L).EQ.J) GO TO 11	564
10	CONTINUE	565
14	K=K+1	566
	GO TO 13	567
	11 IF(L.EQ.K) GO TO 2	568
	KM=K-1	569
	DO 12 LL=L,KM	570
12	ADRD(LL)=AGRD(LL+1)	571
13	ACRD(K)=J	572
2	CENTINUE	573
	IF(III.E0.0) GO TO 3	574
1	CONTINUE	575
3	CONTINUE	576
	WRITE(6,70)	577
70	FCFMAT(1H1.5X. THE ORDER IN WHICH TO CONSIDER THE ACTIVITIES TO DE	578
	*TERMINE THE LUNGEST PATH T(ME: *)	579
	BRITE(6.71)	500
71	FORMAT(1HO.10X.*ORDER ACTIVITY*)	581
	DC 50 I=1.M	582
50	BRITE(6.51) I.AORD(I)	583
51	FGRMAT(1H +10X+15+5X+15)	584
	RETURN	565
	END	586
	FUNCTION CPTIME(CPATHT)	587
C	DETERMINE THE CRITICAL PATH TIME: CPTIME	588
C	XNODE(I) = EARLIEST TIME THAT AN ACTIVITY BEGINNING AT NODE I	589
C	CAN COMMENCE	590
C		591
	IMPLICIT REAL+8 (A-H.O-Z)	592
c .		593
	CCMMON /BLKA/CTIME.DTIME.M	594
	DIMENSION CTIME(0500), DTIME(0500)	595
c .		596
	COMMON /BLAA/XNODE.TAIL.HEAD.AGRO.NMM	597
	INTEGER ADRD(500).TAIL(500).HEAD(500)	598
	DIMENSION XNODE(500)	599
c .		600
	DC 1 I=1.NMM	601
1	XNODE(1)=0.D0	602
	DG 2 II=1.M	603
	I=ACRD(II)	604
2	XNODE(HEAD([])=DMAX1(XNODE(TAIL([])+CTIME([],XNODE(HEAD([])))	605
	[=HEAD(ACRD(M))	606
	CFTIME=XNODE(I)	607
	RETURN	608
	ENO	609

				610
C				611
C				612
C				613
C				614
				615
C	•	•		616
				617
				618
C	•	•		619
				620
				621
				622
C	•	•		623
				625
1				626
				627
2				628
•				629
				630
			그렇게 하는데 어려면 전혀 하게 하는 것이 없는데 되었다면 하는데 그는데 하는데 그렇게 되었다면 되었다면 하는데 그렇게 되었다면 하는데 그를 되었다면 하다.	631
				632
				633
c				634
c			SUBROLTINE FPATH FINDS THE CRITICAL PATH	635
c				636
c				637
			IMPLICIT REAL*8 (A-H.O-Z)	638
			COMMEN /BLKA/CTIME.DTIME.M	639
			DIMENSION CTIME(0500).DTIME(0500)	640
C				641
			COMMON /BLAA/XNODE.TAIL.HEAD.AORD.NMM	642
			INTEGER ACRD(500).TAIL(500).HEAD(500)	643
			DIMENSION XNODE(500)	644
C	•	•		645
			CCMMCN /BLKB/IBB,KB,KKB,KKK	646
			DIMENSION 188(50).KB(50)	647
C	•	•	이 사이들은 사람들이 있다면 하는데 그렇게 되었다면 하는데 되었다면 하는데 그렇게 하는데 그렇게 되었다면 하는데 사람들이 되었다면 하는데 살아 없다면 하는데 그렇게 되었다면서 그 사람들이 되었다면 하는데 그렇게 되었다면 하는데 그렇게 되었다면 하는데 그렇게 되었다면 하는데 그렇게 되었다면서 그렇게 되었다면 하는데 그렇게 되었다면 하는데 그렇게 되었다면서 그렇게 그렇게 되었다면서 그렇게 되었다면서 그렇게 되었다면서 그렇게 되었다면서 그렇게 되었다면서 그렇게 되었다면서 그렇게 그렇게 되었다면서 그렇게 되었다면서 그렇게 되었다면서 그렇게 되었다면서 그렇게 되었다면서 그렇게 되었다면서 그렇게 그렇게 되었다면서 그렇게 되었다면서 그렇게	648
C				649
C				650
C				651
C				652
C				653
C				654
C				655
C				656
-				657
C				658
C				659
•				661
				662
	0	11		663
03	-	•		664
				665
				666
				667
				668
				669
			SMIN=SLACK	670

	ISMIN=1
83000	CONTINUE
83003	IBB(KKK)=ISMIN
	KKK=KKK+1
	KB(KKK)=TAIL(ISMIN)
	IF(TAIL(ISMIN).GT.1) GO TO 83001
	KKB=KKK-1
	RETURN
	END
	SUBROUTINE FINDS
:	
c	SUBROUTINE FINDE EVALUATES THE SUBNETWORK'S DISCRETE DURATION
C	DISTRIBUTION BY EXPLICITLY EVALUATING ALL OF THE SUBNETWORK'S
c	ACTIVITY DURATION CONFIGURATIONS
c	
	IMPLICIT REAL*8(A-H.O-Z)
	COMMON /BLKA/CTIME.DTIME.M
	DIMENSION CTIME(0500).DTIME(0500)
	COMMON /BLKE/INCLUS.NINCL.NCLUS
	DIMENSION HINCL(50). INCLUS(50.100)
	CCMMON /BLKG/FD.ULEFD.LLEFD
	REAL+8 FD(20).ULEFD(20).LLEFD(20)
	CGMMON /BLKI/IEDF.NSUB.SAMSIZ
	INTEGER SAMSIZ
	COMMON /BLKK/IR.18
	CCMMON /BLKS/SUM. ICLGIB
	CCMMGN /BLKY/FDUHAT.FDLHAT.FHAT
	DIMENSION .FOLHAT (3,20), FDUHAT (3,20), FHAT (20)
	1R=1
	• • • • • • • • • • • • • • • • • • • •
c	
c	FORM A CLUSTER CONTAINING ALL OF THE ACTIVITIES IN THE
c	SUBNETWORK
2	
	• • • • • • • • • • • • • • • • • • • •
	NINCL(1)=M
	00 1501 I=1.M
1501	INCLUS(1.1)=I
	00 7101 I=1.IEDF
7101	LLEFD(1)=0.D0
	IB=2**M

c	
	CALL EXACTF

	FHAT(1)=LLEFD(1)/SUM
	DO 6900 1=2.1EDF
	16=1-1
	LLEFO(1)=LLEFO(11)+LLEFD(1)
	FHAT(1)=LLEFD(1)/SUM
6900	CONTINUE
	WRITE (6.6362)

6362	FCHMAT (1H1)
100	FORMAT (1H1. 5x. THE DISCRETE DURATION DISTRIBUTION FOR SUBNETWORK
	••.12)
	NRITE(6,101)
101	FORMAT(1HO.14X. TIME .16X. DISTRIBUTION /)
	DG 102 I=1.1EDF
102	MRITE(6.103) FD(1).FHAT(1)
103	FORMAT(1H .10X.2(E15.5.5X))
	RETURN
	END
	SUBRCUTINE EXACTF
c	SUBROUTINE EXACTE DETERMINES THE EMPIRICAL DISTRIBUTION
c c	FUNCTION FOR THE SUBNETWORK DURATIONS CORRESPONDING TO ALL POSSIBLE ACTIVITY DURATION CONFIGURATIONS
	IMPLICIT REAL+8(A-H.Q-Z)
с	COMMGN /BLKE/INCLUS.NINCL.NCLUS DIMENSIUN NINCL(50).INCLUS(50.100)
c	
	COMMON /BLKG/FD. ULEFD.LLEFD
	REAL +8 FC(20), ULEFD(20), LLEFD(20)
c	
	CGMMON /BLKK/IR.IB
c	COMMON /BLKS/SUM.ICLGIB
c	
	I J=0
	NIB=0
	KJ=NINCL(IR)
200.	SUM= C.DO
2001	NIB=NIB+1
	IF(NIB.GT.IB) GO TO 2000 CALL CCNVRT(NIB.IJ.KJ.SPROB)
	XL=CLTIME(CPATHT)
	X=XL-1.D-10
	1=0
6420	1=1+1
	1F(x.GT.FD(1)) GO TO 6420
	LLEFD(1) = LLEFD(1) + SPROB
	SLM=SUM+SPROB
	GC TO 2001
2000	CONTINUE
	NIB=NIB-1
	RETURN END
	SUBROUTINE CLUSTR
c	OUDIOVITAL GEOSIA
č	SUBROUTINE CLUSTER DETERMINES THE CLUSTER FOR GIVEN THETA AND
č	LAMEDA
c	
c	
	IMPLICIT REAL+8 (A-H.O-Z)
	COMMCN /BLKA/CTIME.DTIME.M
	DIMENSION CTIME(0500).DTIME(0500)
c	
	COMMCN /BLKB/IBB.KB.KKB.KKK
	DIMENSION 188(50) + K8(50)

	COMMON /BLKC/ICRITP.KCPB DIMENSION [CRITP(50)]	793 794
c		795
	CONHCN /BLKD/SIGMA.COT	796
	DIMENSION COT(500) SIGMA(500)	797
c		798
	COMMON /BLKE/INCLUS.NINCL.NCLUS	799
	DIMENSION NINCL(50).INCLUS(50,100)	800
c		801
	COMMON /BLKJ/THETA.LAMBDA	802
_	REAL*8 LAMBDA.THETA	803
c		804
	DIMENSION LEFT(500).LEFTO(500).NONCP(500).NINAG(500) INTEGER ASSGRP(50.50).CLINCL(50.50).EGRP(50).NCLINC(50)	805
c	THIESER ASSURES, SOUTH THE THE THE THE THE THE THE THE THE T	807
c · ·		808
c	THE ASSOCIATE GROUPS ARE NOW FORMED	809
c		810
	WRITE(6.3165)LAMBDA	811
3165	FORMAT(1H1.5X. THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBDA =	812
	*E15.5)	813
	11111=1	814
	DO 2825 I=1.M	815
2825	CTIME(I)=COT(I)	816
	[wwwG=ICRITP(1)	817
2801	CONTINUE	818
22825	CTIME(IWWWQ)=COT(IWWWQ)	819
	Immwg=ICRIYP(XIIII)	820
	TEX=COT(1wwwQ)-LAMBDA*SIGMA(IwwwQ)	821
	IF(TEX.LT.0.D0)TEX=0.D0	822
	CTIME(IWWWQ)=TEX	823
	DUMMY=CPTIME(CPATHT) CALL FPATH	824 825
c	CALL FPAIN	826
c	DETERMINE ASSOCIATE GROUP	827
c	SEPERATE ASSOCIATE SING	828
2910	NINAG(11111)=0	829
	DC 2911 K=1,KKB	830
	KK=1	831
2913	IF(188(K).EQ.ICRITP(KK)) GO TO 2911	832
	IF(KK.GE.KCPB) GO TO 2912	833
	KK=KK+1	834
	GG TO 2913	835
2912	NINAG(IIIII)=NINAG(IIIII)+1	836
	ASSGRP(IIIII.NINAG(IIIII))=188(K)	837
2911	CONTINUE	838
2016	WRITE(6.2915) IIIII.ICRITP(IIIII), NINAG(IIIII)	839
	FORMAT(1H0.10X, THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE ".13. **-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY ".13.", IS = ".13)	841
	IDUCK=NINAG(IIIII)	842
	IF (IDUCK-EQ-0) GO TO 2810	843
	WRITE(6,2916) (ASSGRP(IIIIII,I),I=1,IDUCK)	844
2916		845
	*WS',/,15X,50(13,','))	846
2810	11111=11111+1	847
	IF(11111-LE-KCPB) GO TO 2801	848
c		849
C	DETERMINE THE CLUSTERS	850
c		851
C	THE CLUSTERS ARE POOLED TOWARD THE TERMINAL NODE	852
C	NCLUS = THE NUMBER OF NON-EMPTY CLUSTERS	853

```
NINCL(I) = THE NUMBER OF ACTIVITIES IN THE I-TH CLUSTER
                                                                                 854
               INCLUS(1. J) = THE J-TH ACTIVITY IN THE 1-TH CLUSTER
                                                                                 855
C
               NCLINC(1) = THE NUMBER OF CLUSTERS COMPRISING THE I-TH
                                                                                 856
C
                    CLUSTER AFTER POOLING
C
               CLINCL(I.J) = THE J-TH CLUSTER WHICH HAS BEEN POOLED INTO
                                                                                 858
c
                    THE 1-TH CLUSTER
                                                                                 859
C
                                                                                 860
               NCLINC AND CLINCL HELP KEEP TRACK OF WHICH CLUSTER THE
c
                                                                                 861
c
               CRITICAL PATH ACTIVITIES ARE IN
                                                                                 862
c
                                                                                 863
C
                                                                                  864
C
         BELOW FORMS CLUSTERS BY PUTTING EACH CRITICAL PATH ACTIVITY IN
                                                                                  865
         SEPARATE CLUSTER AND THEN ADDING EACH CRITICAL PATH ACTIVITY'S
                                                                                  866
C
c
         ASSUCIATES TO ITS CLUSTER
                                                                                  867
C
                                                                                  868
      NCLUS=KCPB
                                                                                  869
      DO 3020 I=1.KCPB
                                                                                  870
      NCLINC(I)=1
                                                                                  871
      CLINCL(I.1)=1
                                                                                  872
                                                                                 873
      MINCL(I) = MINAG(I)+1
      INCLUS(1.1)=ICRITP(1)
                                                                                  874
                                                                                 875
      IF(NINAG(1).EQ.0) GO TO 3020
      IDUCK=NINCL(1)
                                                                                 876
      DO 3021 J=2. IDUCK
                                                                                  877
                                                                                 878
      1-1-1
3021
     INCLUS(I.J)=ASSGRP(I.JJ)
                                                                                  879
3020
      CONTINUE
                                                                                 880
C
                                                                                  881
C
         BELOW POOLS CLUSTERS FORMED FROM ASSOCIATES
                                                                                 AA2
                                                                                  883
C
      IA=0
                                                                                 884
                                                                                  885
3031
     IA=IA+1
      IF(IA.GE.KCPB) GO TO 3030
                                                                                  886
      IF(NCLUS.EQ.1) GO TO 3030
                                                                                  887
      IDIA=NINCL(IA)
                                                                                  888
      IF(IDIA-EQ-0) GO TO 3031
                                                                                 889
                                                                                 890
      IAA=IA+1
      DO 3023 11=1AA.KCPB
                                                                                  891
      IDII=NINCL(II)
                                                                                 892
      IF(ICII.EQ.0) GO TO 3023
                                                                                  893
      DO 3025 I=1.IDIA
                                                                                 894
      CO 3025 J=1.1011
                                                                                  895
      IF(INCLUS(II, J).EQ.INCLUS(IA.I)) GO TO 3027
                                                                                 896
3025 CENTINUE
                                                                                 897
      GC TO 3023
                                                                                  898
      NCLUS=NCLUS-1
3027
                                                                                 ROO
      DO 3028 J=1.IDII
                                                                                  900
      DC 3029 I=1.1DIA
                                                                                  901
      IF(INCLUS(II, J).EQ. INCLUS(IA.I)) GO TO 3028
                                                                                  902
      CONTINUE
                                                                                  903
      NINCL(IA)=NINCL(IA)+1
                                                                                  904
      INCLUS(IA.NINCL(IA) )=INCLUS(II.J)
                                                                                  905
3028 CONTINUE
                                                                                  906
      NINCL(II)=0
                                                                                  907
      NCLINC(IA)=NCLINC(IA)+1
                                                                                  908
      CLINCL(IA.NCLINC(IA)) = II
                                                                                  909
      NCLINC(11)=0
                                                                                  910
3023 CONTINUE
                                                                                  911
      GC TC 3031
                                                                                  912
      CONTINUE
                                                                                  913
3030
                                                                                  914
```

```
BELCW DESCRIBES CLUSTERS AFTER POCLING BASED ON THE ASSOCIATES
C
                                                                                 916
      ##11E(6.3033) NCLUS
                                                                                 917
3033 FORMAT(1HG./6x. THERE ARE . 13. NONEMPTY CLUSTERS AFTER POOLING C
                                                                                 918
     AN THE BASIS OF ASSOCIATES ONLY. ..
                                                                                 919
      11=0
                                                                                 920
      DO 3034 1=1.KCPB
                                                                                 921
      IF(NINCL(1).EQ.0) GO TO 3034
                                                                                 922
      11=11+1
                                                                                 923
      IDUCJ=NINCL(I)
                                                                                 924
      #RITE(6.3035) I.(INCLUS(1.J).J=1.IDUCJ)
                                                                                 925
3035 FORMAT (1HO. LOX. THE ACTIVITIES IN THE . 13. -TH CLUSTER ARE AS FOL
                                                                                 926
     *LC#5: ./.15x.50(13, ...))
                                                                                 927
3034 CONTINUE
                                                                                 928
      DC 3101 I=1.M
                                                                                 929
      LEFT(1)=0
                                                                                 930
      00 3102 J=1.KCPB
                                                                                 931
      IF(NINCL(J).EQ.0) GO TO 3102
                                                                                 932
      IBUCK=NINCL(J)
                                                                                 933
      DO 3110 K=1.IDUCK
                                                                                 934
      IF(I.EQ.INCLUS(J.K)) GO TO 3107
                                                                                 935
3110
      CCNTINLE
                                                                                 936
      CONTINUE
3102
                                                                                 937
      GC TG 3101
                                                                                 938
      LEFT(I)=J
3107
                                                                                 939
3101
     CONT INUE
                                                                                 940
C
                                                                                 941
C
         LEFTOVERS ARE ACTIVITIES NOT IN CLUSTERS AFTER ASSOCIATES HAVE
                                                                                 942
         BEEN CONSIDERED BUT BEFORE ELIMINANTS HAVE BEEN CONSIDERED
C
                                                                                 943
C
                                                                                 944
C
                                                                                 945
C
         DETERMINE THE NUMBER OF LEFTOVERS. NLEFT
                                                                                 946
C
              LEFTO(L) = J IMPLIES THAT THE L-TH LEFTCVER IS THE J-TH
                                                                                 947
C
                             ACTIVITY
                                                                                 Q4 A
                                                                                 949
      NLEFT=0
                                                                                 950
      DO 3122 J=1.M
                                                                                 951
      IF(LEFT(J).NE.0) GO TO 3122
                                                                                 952
      NLEFT=NLEFT+1
                                                                                 953
      LEFTO(NLEFT)=J
                                                                                 954
3122 CCATINUE
                                                                                 955
      WRITE(6.3123) NLEFT
                                                                                 956
3123 FORMAT (1HO.10X. THERE ARE .. I3. ACTIVITIES NOT IN ANY CLUSTER YE
                                                                                 957
     *T.")
                                                                                 958
      WRITE(6,3323)THETA
                                                                                 959
3323 FORMAT(1H1,5x, THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY A
                                                                                 960
     *RE NOW DETERMINED FOR THETA = 1.E15.5)
                                                                                 961
C
                                                                                 962
C
         ELIMINANTS FOR EACH NGN-CRITICAL-PATH ACTIVITY ARE NOW
                                                                                 963
C
         DETERMINED
                                                                                 964
C
              NNNCP = THE NUMBER OF ACTIVITIES NOT ON THE CRITICAL PATH
                                                                                 965
              NONCP(LE) = THE LE-TH ACTIVITY NOT ON THE CRITICAL PATH
C
                                                                                 966
C
                                                                                 967
      NAACP=M-KCP8
                                                                                 968
      LE=0
                                                                                 969
      DO 5000 1=1.M
                                                                                 970
      J= 1
                                                                                971
      IF(I.EQ.ICRITP(J)) GC TO 5000
                                                                                 972
                                                                                 973
      IF(J.LE.KCPU) GO TO 5001
                                                                                974
5002 LE=LE+1
                                                                                 975
```

```
976
      NCNCP(I E)=1
                                                                                 977
5000 CCNTINUE
      WRITE(6.5005) NNNCP
                                                                                  978
5005 FORMAT(1HO.10x. THERE ARE .. 13. ACTIVITIES NOT ON THE CRITICAL PA
                                                                                 979
     *TH. THEY ARE AS FOLLOWS: 1)
                                                                                 980
      IF (NNNCP.EQ.0) GO TO 3124
                                                                                 981
      00 5006 I=1.LE
                                                                                 982
5006
      WRITE(6.5007) 1.NONCP(1)
                                                                                 FRP
5007 FCRMAT(1H .15X.13.'. '.13)
                                                                                 984
      LE=0
                                                                                 985
3126 LE=LE+1
                                                                                 986
      CTIME(IWWWG) = COT(IWWWG)
                                                                                 987
      CTIME(NONCP(LE)) = CGT(NONCP(LE)) + THETA+SIGMA(NONCP(LE))
                                                                                 988
 7756 INWWG = NCNCP(LE)
                                                                                 989
      DUMMY=CPT [ME(CPATHT)
                                                                                 990
      CALL FRATH
                                                                                  991
3121 CONTINUE
                                                                                 992
C
                                                                                 993
C
         DETERMINE THE ELIMINANTS OF THE LE-TH ACTIVITY NOT ON THE
                                                                                 994
C
         CRITICAL PATH
                                                                                 995
C
                         = THE NUMBER OF ELIMINANTS FOR THE LE-TH
              NE
                                                                                 996
C
                           ACTIVITY NOT ON THE CRITICAL PATH
                                                                                 997
                            = THE J-TH ELIMINANT FOR THE LE-TH ACTIVITY
C
              FGRP(1)
                                                                                 RPP
C
                              NOT ON THE CRITICAL PATH
                                                                                 999
C
                                                                                1000
      NE=0
                                                                                1001
      DO 3130 K=1.KCPB
                                                                                1002
      DO 3131 I=1.KKB
                                                                                1003
      IF (188(1) . EQ. (CRITP(K)) GO TO 3130
3131 CCATINUE
                                                                                1005
      NE=NE+1
                                                                                1006
      EGRP(NE)=ICHITP(K)
                                                                                1007
3130
     CENTINUE
                                                                                1008
      WRITE(6.3133) NE.NONCP(LE)
                                                                                1009
3133 FORMAT(1HO.10X. THERE ARE .. 13. ELIMINANTS CORRESPONDING TO ACTIV
                                                                                1010
     *ITY '.13)
                                                                                1012
      IF (NE.EQ.O) GO TO 3171
                                                                                1012
      DO 3135 K=1.NE
                                                                                1013
3135 WRITE(6.3136) K.NONCP(LE).EGRP(K)
                                                                                1014
3136 FORMAT(1H .14X. THE .. 13. -TH ELIMINANT CORRESPONDING TO ACTIVITY
                                                                                1015
     * '. 13. ' IS ACTIVITY '.13)
C
                                                                                1017
c
         DETERMINE WHETHER NONCP(LE) IS AN ASSOCIATE
                                                                                1018
              JA = 1 IF NGNCP(LE) IS AN ASSOCIATE

JA = 2 IF NGNCP(LE) IS NOT AN ASSOCIATE
C
                                                                                1019
C
                                                                                1020
C
      K=NCNCP(LE)
                                                                                1022
      JA=1
      IF (LEFT (K) . EQ.O) JA=2
                                                                                1024
      IF (JA.EG.2) GO TO 5010
      IT=LEFT(K)
                                                                                1026
C
                                                                                1027
C
         THE IT-TH CLUSTER IS EXPANDED TO INCLUDE ELIMINANTS
                                                                                1028
C
                                                                                1029
      GC TC 5011
                                                                                1030
5010
     CONTINUE
                                                                                1031
                                                                                1032
         ITTIT IS THE ACTIVITY NUMBER OF THE FIRST ELIMINANT
C
                                                                                1033
C
         IT IS THE CLUSTER TO WHICH THE FIRST ELIMINANT CURRENTLY BELONG
                                                                                1034
C
                                                                                1035
      ITTT=EGRP(1)
                                                                                1036
```

	[I=LEFT(ITIT) LEFT(NONCP(LE))=IT	103
c	ELF FRANCE (ELF)-11	103
č	THE IT-TH CLUSTER IS EXPANDED TO INCLUDE ELIMINANTS	104
č	THE TI-TH CEOSTER IS EXPANDED TO INCLUDE ELIPINATIS	104
•	NINCL(IT)=NINCL(IT)+I	104
	INCLUS(IT, NINCL(IT)) = NONCP(LE)	104
	IF(NE.EQ.1) GO TO 3171	104
5011	DO 3172 J=JA.NE	104
C	THE THE ACTIVITY NUMBER OF THE MENT OF THE MANAGE	104
C	IU IS THE ACTIVITY NUMBER OF THE NEXT ELIMINANT	104
C	IF IU IS IN CLUSTER K. THEN CLUSTER K IS POOLED INTO CLUSTER IT	104
C		104
	IU=EGRP(J)	105
	K=LEFT(IU)	105
3182	1F(17.EQ.K) GO TO 3172	105
	NCLUS=NCLUS-1	105
	IW=NCLINC(K)	105
	DO 3183 IA=1.IW	105
	LEFT(ICRITP(CLINCL(K,IA)))=IT	105
	NCLINC(IT)=NCLINC(IT)+1	105
3183	CLINCL(IT.NCLINC(IT))=CLINCL(K.IA)	105
	ACLINC(K)=0	105
	IW=NINCL(K)	106
	NINCL(K)=0	106
	DO 3184 IA=1.1W	106
	LEFT(INCLUS(K, IA))=IT	106
	AIACL(IT)=NINCL(IT)+1	106
3184	INCLUS(IT.NINCL(IT))=INCLUS(K.IA)	106
3172	CONTINUE	106
3171	CONTINUE	106
	IF(LE.LT.NNNCP) GO TO 3126	106
C		106
c	END OF PLULING BASED ON ELIMINANTS EXCEPT FOR THE FOLLOWING	107
C	DESCRIPTION	107
c		107
	WRITE(6.3173) NCLUS	107
3173		107
	ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.')	107
	DO 3176 1=1,KCPB	107
	IF(NINCL(1).EQ.0) GO TO 3176	107
	IDD=NINCL(I)	107
	WRITE(6.3174) NINCL(1).1.([NCLUS(1.J).J=1.1DD)	107
3174		108
3174		108
	•ER. THEY ARE AS FOLLOWS: • ,/ .20X .50([3. • ,*])	
	IDUCK=NCLINC(I)	108
	WRITE(6.3175) NCLINC(1).(CLINCL(1.J).J=1.IDUCK)	108
3175	FORMAT (1HO.15X.13. CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER	108
	*. THEY WERE AS FOLLOWS: './.20x.50(13. '. '))	108
	CGNTINUE	108
3124	CCNTINUE	108
	RETURN	108
	END	108
	SUBROUTINE UNION	109
C		109
C	UNION DETERMINES UPPER AND LOWER BOUNDS ON	109
C	THE DISTRIBUTION FUNCTION OF THE SUBNETWORK DURATION	109
c	BASED ON THE UNION OF THE CLUSTERS	109
c		109
11-11	IMPLICIT REAL+8(A-H.O-Z)	109
c		109

	COMMCN /BLKA/CTIME.DTIME.M DIMENSION CTIME(0500).DTIME(0500)	10
c		11
	CCMMON /BLKC/ICRITP.KCPB	11
	DIMENSION ICRITP(50)	11
c		11
	COMMON /BLKE/INCLUS.NINCL.NCLUS	11
	DIMENSION NINCL(50), INCLUS(50,100)	1 1
c		11
	COMMON /BLKG/FD.ULEFD.LLEFD	11
_	REAL+8 FD(20).ULEFD(20).LLEFD(20)	11
c		11
	CCMMON /BLKH/FHI.FLQ.PP.PQ GIMENSION FHI(500).FLQ(500).PP(500).PQ(500)	11
c	CIPERSION FRICSOOTIFECTSOOTIFCTSOOT	11
	CCMMON /BLKI/IEDF.NSUB.SAMSIZ	11
	INTEGER SAMSIZ	11
c		11
	CCMMCN /BLKK/IR.IB	11
c		11
	CCMMON /BLKL/FLHAT.FUHAT	11
	DIMENSION FLHAT(20).FUHAT(20)	11
c		11
	CCMMGN /BLKS/SUM.ICLGIB	1 1
c	•••••••••••	11
	IF(NCLUS.GT.1) GO TO 6000	11
	I=0	11
6001	1=1+1	11
	1F(NINCL(1).EQ.0) GO TO 6001	11
	NCL=1	11
	GO TO 6002	11
c••••		
	GO TO 6002	11
C**** C C	GG TG 6002	11
c	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER	11
c c	POGL ALL OF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER	111
c c c	POGL ALL OF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER	111
C	POGL ALL OF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POGLED CLUSTER NMAX = 0	111111111111111111111111111111111111111
C C C	POGL ALL OF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POGLED CLUSTER NMAX = 0 DO 6003 1=1,KCPB	111111111111111111111111111111111111111
C C C	POGL ALL OF THE PREVIOUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POGLED CLUSTER NMAX = 0 DO 6003 I=1, KCPB IF(NINCL(1).LE.NMAX) GO TO 6003	111111111111111111111111111111111111111
C C C	GG TO 6002 POOL ALL UF THE PREVIOUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POOLED CLUSTER NMAX = 0 DO 6003 I=1.KCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL =I	111111111111111111111111111111111111111
C C C C	GG TO 6002 POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1.KCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL =I NMAX=NINCL(I)	111111111111111111111111111111111111111
C C C C	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1.kCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL =I NMAX=NINCL(I) CONTINUE	111111111111111111111111111111111111111
C C C C****	POGL ALL UF THE PREVIOUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1.kCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL =I NPAX=NINCL(I) CONTINUE DO 6004 I=1.kCPB	
C C C C	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1.KCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL = I NPAX=NINCL(I) CONTINUE DO 6004 I=1.KCPB IF(NINCL(I).EQ.0) GO TO 6004	
C C C C****	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1.KCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL = I NPAX=NINCL(I) CONTINUE DO 6004 I=1.KCPB IF(NINCL(I).EQ.0) GO TO 6004 IF(I.EQ.NCL) GO TO 6004	
C C C C	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1.kCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL = I NPAX=NINCL(I) CONTINUE DO 6004 I=1.kCPB IF(NINCL(I).EQ.0) GO TO 6004 IF(I.EQ.NCL) GO TO 6004 K=NINCL(NCL)	
C C C C****	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1.KCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL =I NMAX=NINCL(I) CONTINUE DO 6004 I=1.KCPB IF(NINCL(I).EQ.0) GO TO 6004 IF(I.EQ.NCL) GO TO 6004 K=NINCL(NCL) JJ=NINCL(I)	
C ****	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1.kCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL = I NPAX=NINCL(I) CONTINUE DO 6004 I=1.kCPB IF(NINCL(I).EQ.0) GO TO 6004 IF(I.EQ.NCL) GO TO 6004 K=NINCL(NCL)	
C ****	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POGLED CLUSTER NMAX = 0 DO 6003 l=1.KCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL =I NMAX=NINCL(I) CONTINUE DO 6004 I=1.KCPB IF(NINCL(I).EQ.0) GO TO 6004 IF(1.EQ.NCL) GO TO 6004 K=NINCL(NCL) JJ=NINCL(I) NINCL(I)=0	
C	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POGLED CLUSTER NMAX = 0 DO 6003 1=1.KCPB IF(NINCL(1).LE.NMAX) GO TO 6003 NCL =I NMAX=NINCL(1) CONTINUE DO 6004 1=1.KCPB IF(NINCL(1).EQ.0) GO TO 6004 IF(I.EQ.NCL) GO TO 6004 K=NINCL(NCL) JJ=NINCL(I) NINCL(I)=0 DO 6005 J=1.JJ	
C	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1.KCPB IF (NINCL(I).LE.NMAX) GO TO 6003 NCL = I NMAX=NINCL(I) CONTINUE DO 6004 I=1.KCPB IF (NINCL(I).EQ.0) GO TO 6004 IF (I.EQ.NCL) GO TO 6004 K=NINCL(NCL) J=NINCL(I) NINCL(I)=0 DO 6005 J=1.JJ K=K+1	
5003	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POGLED CLUSTER NMAX = 0 DO 6003 1=1.KCPB IF (NINCL(I).LE.NMAX) GO TO 6003 NCL = I NMAX=NINCL(I) CONTINUE DO 6004 1=1.KCPB IF (NINCL(I).EQ.0) GO TO 6004 IF (I.EQ.NCL) GO TO 6004 K=NINCL(NCL) JJ=NINCL(I) NINCL(I)=0 DO 6005 J=1.JJ K=K+1 INCLUS(NCL.K)=INCLUS(I.J)	
6005	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS PCGLED CLUSTER NMAX = 0 DO 6003 I=1,KCPB IF (NINCL(I).LE.NMAX) GO TO 6003 NCL = I NMAX=NINCL(I) CONTINUE DO 6004 I=1.KCPB IF (NINCL(I).EQ.0) GO TO 6004 K=NINCL(NCL) J=NINCL(I) NINCL(I)=0 DO 6005 J=1.JJ K=K+1 INCLUS(NCL.K)=INCLUS(I.J) NINCL(NCL)=NINCL(NCL)+JJ	
6003	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POGLED CLUSTER NMAX = 0 DO 6003 I=1,KCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL =I MAX=NINCL(I) CONTINUE DO 6004 I=1.KCPB IF(NINCL(I).EQ.0) GO TO 6004 IF(I.EQ.NCL) GO TO 6004 K=NINCL(NCL) JJ=NINCL(I) NINCL(I)=0 DO 6005 J=1.JJ K=K+1 INCLUS(NCL.K)=INCLUS(I.J) NINCL(NCL)=NINCL(NCL)+JJ CONTINUE NNCL=NINCL(NGL)	
6003	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POGLED CLUSTER NMAX = 0 DO 6003 I=1,KCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL = I NMAX=NINCL(I) CONTINUE DO 6004 I=1.KCPB IF(NINCL(I).EQ.0) GO TO 6004 IF(I.EQ.NCL) GO TO 6004 IF(I.EQ.NCL) GO TO 6004 K=NINCL(NCL) J=NINCL(I) NINCL(I)=0 DO 6005 J=1.JJ K=K+1 INCLUS(NCL.K)=INCLUS(I.J) NINCL(NCL)=NINCL(NCL)+JJ CONTINUE NNCL=NINCL(NCL) ICLGIB = -1 IF ALL CONFIGURATIONS ARE TO BE SAMPLED	
c c c	POGL ALL UF THE PREVICUS CLUSTERS INTO ONE CLUSTER NCL = THE INDEX OF THE RESULTANT POOLED CLUSTER NNCL = THE NUMBER OF ACTIVITIES IN THIS POGLED CLUSTER NMAX = 0 DO 6003 I=1,KCPB IF(NINCL(I).LE.NMAX) GO TO 6003 NCL =I MAX=NINCL(I) CONTINUE DO 6004 I=1.KCPB IF(NINCL(I).EQ.0) GO TO 6004 IF(I.EQ.NCL) GO TO 6004 K=NINCL(NCL) JJ=NINCL(I) NINCL(I)=0 DO 6005 J=1.JJ K=K+1 INCLUS(NCL.K)=INCLUS(I.J) NINCL(NCL)=NINCL(NCL)+JJ CONTINUE NNCL=NINCL(NGL)	

c	SIMPLE RANDOM SAMPLING
	IF(NNCL.LT.31) GO TO 1
	ICLGIB=1
	GO TO 2
1	IB=2**NNCL
	ICLGIB=-1
	IF(SAMSIZ.LT.IB) ICLGIB=0
2	CONTINUE
	IR=NCL
C****	***************************************
C	
C	SET ALL ACTIVITITY DURATIONS GUTSIDE THE IR-TH CLUSTER:
c	DISTRIBUTION
c	CTIME() = UPPER PGINT FOR LOWER BOUND ON SUBNETWORK DURATION
c	DTIME() = LOWER POINT FOR UPPER BOUND ON SUBNETWORK DURATION
c	DISTRIBUTION
č	2.311.1001.101
C****	
	00.000
	DO 6006 I=1.M
	DT[ME(1)=FLO(1)
6006	CTIME(1)=FH1(1)
	DO 7101 I=1,1EDF
	ULEFC(1)=0.00
7101	LLEFD(I)=0.D0
C****	***************************************
C	
	CALL BOUND
C	
C	SUBROUTINE BOUND DETERMINES
C	EMPIRICAL DISTRIBUTION FUNCTION FOR THE SUBNETWORK DURATIONS
c	CORRESPONDING TO THE ACTIVITY DURATION CONFIGURATIONS SPECIFIED
c	
C****	***************************************
• • • • • •	FLHAT(1)=LLEFD(1)/SUM
	FUHAT(1)=ULEFD(1)/SUM
	DC 6900 I=2, IEDF
	[1] 그렇게 보고 있었다는 경영구역 프로젝트 (1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.
	11=1-1
	LLEFC(1)=LLEFD(11)+LLEFD(1)
	FLHAT(1)=LLEFD(1)/SUM
	ULEFD(I)=ULEFD(II)+ULEFD(I)
	FUHAT(I)=ULEFD(I)/SUM
6900	CONTINUE
	RETURN
	END
	SUBROUTINE SINGLE
C	
c	SINGLE DETERMINES UPPER AND LOWER BOUNDS ON
C	THE DISTRIBUTION FUNCTION OF THE SUBNETWORK DURATION
c	BASED ON THE INDIVIDUAL CLUSTERS - SEE TECHNICAL REPORT 51
c	The state of the s
	IMPLICIT REAL+B(A-H.O-Z)
c	
	COMMON /BLKA/CTIME.DTIME.M
	DIMENSION CTIME(0500).DTIME(0500)
-	
c	60HH0b 48H64160H70 H600
	COMMON /BLKC/ICRITP.KCPB
4111	DIMENSIGN ICRITP(50)
	CGMMGN /BLKE/INCLUS.NINCL.NCLUS
	DIMENSION NINCL(50). INCLUS(50.100)

c	COMMCN /BLKG/FD.ULEFD.LLEFD
	REAL *8 FD(20).ULEFD(20).LLEFD(20)
c	
	COMMON /BLKH/FHI.FLO.PP.PQ
	DIMENSION FH1(500).FL0(500).PP(500).P0(500)
	COMMON /BLKI/IEDF.NSUB.SAMSIZ
	INTEGER SAMSIZ
	COMMON /BLKK/IR.IB
	COMMON /BLKL/FLHAT.FUHAT
	DIMENSION FLHAT(20).FUHAT(20)
	CCMMON /BLKS/SUM,ICLGIB
	00.0000 1-1.1505
	00 2020 I=1.1EDF
	FLHAT(1)=0.00
020	FUHAT(I)=0.D0
	IR=0
200	IR=IR+1
	IF (IR.GT.KCPB) GO TO 3208
	IF(NINCL(IR).EQ.0) GO TO 3200
	NCL=IR
	NNCL=NINCL(IR)
	ICLGIB = -1 IF ALL CONFIGURATIONS ARE TO BE SAMPLED
	ICLGIB = 0 IF SAMSIZ CONFIGURATIONS ARE TO BE SAMPLED USING
	STRATIFIED RANDOM SAMPLING
	ICLGIB = 1 IF SAMSIZ CONFIGURATIONS ARE TO BE SAMPLED USING
	SIMPLE RANDOM SAMPLING
	SIMPLE KANDUM SAMPLING
	IF(NNCL.LT.31) GO TO 1
	ICLGIB=1
	GO TO 2
	18=2**NCL
	ICLGIB=-1 IE/SANSIZ-LT-18) ICLGIB=0
	IF(SAMSIZ.LT.1B) ICLGIB=0
	CONTINUE
	SET ALL ACTIVITITY DURATIONS OUTSIDE THE IR-TH CLUSTER:
	CTIME() = UPPER POINT FOR LOWER BOUND ON SUBNETWORK DURATION
	DISTRIBUTION
	CTIME() = LOWER POINT FOR UPPER BOUND ON SUBNETWORK DURATION
	DISTRIBUTION

	DO 6006 I=1-M
	CT IME(1)=FLO(1)
006	CTIME(1)=FH1(1)
300	DO 7101 I=1.IEDF
	ULEFD(1)=0.00
101	LLEFD(1)=0.00

••••	
	CALL BOUND
	and the second s
100	SUBROUTINE BOUND DETERMINES
	EMPIRICAL DISTRIBUTION FUNCTION FOR THE SUBNETWORK DURATIONS
	EFF INTERE DISTRIBUTION FUNCTION FUN THE SUBNETBURK DURATIONS

C	CORRESPONDING TO THE ACTIVITY DURATION CONFIGURATIONS SPECIFIED	1:
	***************************************	1
	XXXX=LLEFD(1)/SUM	1:
	IF(XXXX.GT.FLHAT(1)) FLHAT(1)=XXXX	1
	XXXX=ULEFD(1)/SUM	12
	IF(XXXX.LT.FUHAT(1)) FUHAT(1)=XXXX	12
	DC 6900 1=2.1EDF	1.
	11=1-1	12
	LLEFO(I)=LLEFO(II)+LLEFO(I)	12
	XXXX=LLEFD(1)/SUM	12
	IF(XXXX.GT.FLHAT(I)) FLHAT(I)=XXXX	12
	ULEFO(I)=ULEFD(II)+ULEFD(I)	12
	XXXX=ULEFD(1)/SUM	12
	IF(XXXX-LT-FUHAT(I)) FUHAT(I)=XXXX	12
6900	CONTINUE	12
	GC TO 3200	12
320e	CGNTINUE	12
	RETURN	12
	END	1.
	SUBROUTINE BOUND	1:
c		1.
c	SUBROUTINE BOUND DETERMINES THE	1
č	EMPIRICAL DISTRIBUTION FUNCTION FOR THE SUBNETWORK DURATIONS	1
č	CORRESPONDING TO THE ACTIVITY DURATION CONFIGURATIONS SPECIFIED	1
c	CONTESPONDING TO THE ACTIVITY BONNIES CONTIGUES SPECIFIES	1
c	ALL ACTIVITIES OUTSIDE THE CLUSTER BEING CONSIDERED HAVE	1
č	FIXED DURATIONS	-
		1
c	ACTIVITIES INSIDE THE CLUSTER BEING CONSIDERED HAVE BOTH	1.
c	UPPER AND LOWER DURATIONS CONSIDERED.	1
C	EITHER ALL COMBINATIONS OF UPPER AND LOWER POINTS ARE	1
C	CONSIDERED OR A RANDOM SAMPLE THEREOF.	1.
c		1:
	IMPLICIT REAL+8(A-H.C-Z)	1.
c		1 .
	COMMON /BLKG/FD.ULEFD.LLEFD	1 :
	REAL+8 FD(20).ULEFD(20).LLEFD(20)	1 :
c		1:
	CGMMUN /BLKI/IEDF.NSUB.SAMSIZ	1.
	INTEGER SAMSIZ	1
c		1.
	COMMON /BLKK/IR. IB	1
c		1
	COMMON /BLKS/SUM.[CLGIB	1
	CURRON / DENS/30M (ICEUSO	1
c		
	17=13579753	1.
	N1B=0	1
	SUM=0.D0	13
	IF(ICLGIB) 3.6210.6211	1
3	CCNT INUE	1:
C****		1:
C		1 :
C	ALL ACTIVITY DURATION CONFIGURATIONS ARE TO BE CONSIDERED	1.
c		1.
		1.
2001	NIB=NIB+I	1.
	IF(NLB-GT-18) GO TO 2000	1.
	CALL ALL(NIB.XL.XU.SPROB)	1.
	X=XL-1.D-10	1
	1=0	1
6420	1=141	1

	IF(X.CT.FD(1)) GG TG 6420	1342
	LLEFD(I) = LLEFD(I) + SPROB SUM=SUM+SPROB	1343
	X=XU-1.D-10	1345
	1=0	1346
9420	1=1+1	1347
	IF(x.GT.FD(1)) GO TO 9420	1348
	ULEFD(I) = ULEFD(I) + SPROB	1349
	GG 10 2001	1350
6210	CONTINUE	1351
	•••••••••	1352
C	A CTANTICLE CAMPEN CAMPLE OF THE ACTIVITY DUBATION	1353
C	A STRATIFIED RANDOM SAMPLE OF THE ACTIVITY DURATION CONFIGURATIONS IS TO BE CONSIDERED	1354
c	CUNFIGURATIONS IS TO BE CONSIDERED	1356
_		1357
2011	NIB=NIB+1	1358
	IF(NIB.GT.SAMSIZ) GG TG 2000	1359
	CALL STRAT(NId.XL.XU.SPRCB.IY)	1360
	X=XL-1.D-10	1361
	1 = 0	1362
7420	f = I + 1	1363
	IF(X.GT.FD(1)) GO TO 7420	1364
	LLEFD(1) = LLEFD(1) + SPROB	1365
	SUM=SUM+SPROB X=XL-1.D-10	1366
	1=0	1368
74201	1=1+1	1369
	IF(x.GT.FD(1)) GO TO 74201	1370
	ULEFD(1) = ULEFD(1) + SPRQB	1371
	GC TO 2011	1372
6211	CONTINUE	1373
C****	***************************************	1374
C		1375
C	A SIMPLE RANDUM SAMPLE OF THE ACTIVITY DURATION	1376
C	CONFIGURATIONS IS TO BE CONSIDERED	1377
	• • • • • • • • • • • • • • • • • • • •	1378
3011	N18=N18+1	1380
	IF (NIB.GT.SAMSIZ) GO TO 2000	1381
	CALL SIMPLE(NIB, XL, XU, SPROB, IY)	1382
	X=XL-1.D-10	1383
	1=0	1384
8420	1=1+1	1385
	IF(X.GT.FC(I)) GO TO 8420	1386
	LLEFD(I) = LLEFD(I) + SPROB	1387
	SUM=SUM+SPROB	1388
	X=XU-1.D-10 1=0	1389
84201	1=1+1	1390
04201	IF(X.GT.FD(II) GO TO 84201	1392
	ULEFD(I) = ULEFD(I) + SPROB	1393
		1394
2000	CONTINUE	1395
		1396
	RETURN	1397
	END	1398
	SUBROUTINE ALL(NIB.CPTIM.CLTIM.SPROB)	1399
c	ALL RETURNS THE COMPLETION TIME AND PROBABILITY FOR THE	1400
c	ALL RETURNS THE COMPLETION TIME AND PROBABILITY FOR THE NIB-TH CONFIGURATION OF THE 2**NINCL(L) POSSIBLE CONFIGURATIONS	1401
	THE ETHINGELLY PUBLICLE CONFIGURATIONS	1402

		IMPLICIT REAL+8 (A-H.O-Z)
		COMMON / FLKE/INCLUS.NINCL.NCLUS
		DIMENSION NINCL(50).INCLUS(50.100)
	•	COMMON /BLKK/IR. IB
•	•	IJ=0
		사료 취임하는 것이 되고 있었다면 그 아이들이 하는 것이 되었다면 하는 것이 되었다면 하는 것이 되었다면 하는 것이 되었다면 하는데 되었다면 하는데 되었다면 하는데 되었다면 하는데 되었다면 하는데
		KJ=NINCL(IR)
		CALL CONVRT(NIB.1J.KJ.SPROB)
		CPTIM=CPTIME(CPATHT)
		CLTIM=CLTIME(CPATHT)
		RETURN
		END
		SUBFOUTINE CONVRT(JP.IJ.KJ.SPROB)
		IMPLICIT REAL+8 (A-H.O-Z)
		COMMON /BLKA/CTIME.DTIME.M
		DIMENSION CTIME(0500).DTIME(0500)
		COMMON /BLKE/INCLUS.NINCL.NCLUS
		DIMENSION NINCL(50). INCLUS(50.100)
		COMMON /BLKH/FHI.FLO.PP.PQ
		DIMENSION FHI (500) .FLO(500) .PP(500) .PQ(500)
	•	COMMON /BLKK/IR, IB
•	•	
		CONVERT THE INTEGER JP TO A BINARY NUMBER TO DEFINE THIS
		SECHENT OF THE ACTIVITY CONFICINATION
		SEGMENT OF THE ACTIVITY CONFIGURATION
		SPROB=1.00
		SPROB=1.00 K=JP
		SPROB=1.00 K=JP DO 1 I=1.KJ
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR,I+IJ)
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR,I+IJ)
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR,I+IJ) CTIME(L)=IZ+FHI(L)+(1-IZ)+FLO(L)
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ+FHI(L)+(1-IZ)+FLO(L) DTIME(L)=CTIME(L)
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ+FHI(L)+(1-IZ)+FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ+PQ(L)+(1-IZ)+PP(L))
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ+FHI(L)+(1-IZ)+FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ+PQ(L)+(1-IZ)+PP(L)) K=IH
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ+FHI(L)+(1-IZ)+FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ+PQ(L)+(1-IZ)+PP(L)) K=IH RETURN
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ+FHI(L)+(1-IZ)+FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ+PQ(L)+(1-IZ)+PP(L)) K=IH RETURN END
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ+FHI(L)+(1-IZ)+FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ+PQ(L)+(1-IZ)+PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY)
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 2**NINCL(IR)
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 2**NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 2**NINCL(IR)
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR,I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 20*NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 20*NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION THE ROUTINE ASSUMES THAT A SAMPLE OF SIZE SAMSIZ WILL BE
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR,I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 20*NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ+FHI(L)+(1-IZ)+FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ+PQ(L)+(1-IZ)+PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 20+NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION THE ROUTINE ASSUMES THAT A SAMPLE OF SIZE SAMSIZ WILL BE GENERATED AND RETURNS ONLY ONE SAMFLE POINT PER CALL
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB*(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB*CPTIM*CLTIM*SPROB*IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 2**NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION THE ROUTINE ASSUMES THAT A SAMPLE OF SIZE SAMSIZ WILL BE GENERATED AND RETURNS ONLY ONE SAMFLE POINT PER CALL IMPLICIT REAL*8(A-H*O-Z)
		SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 20*NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION THE ROUTINE ASSUMES THAT A SAMPLE OF SIZE SAMSIZ WILL BE GENERATED AND RETURNS ONLY ONE SAMFLE POINT PER CALL IMPLICIT REAL*8(A-H.O-Z)
	•	SPROB=1.00 K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB*(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 2**NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION THE ROUTINE ASSUMES THAT A SAMPLE OF SIZE SAMSIZ WILL BE GENERATED AND RETURNS ONLY ONE SAMFLE POINT PER CALL IMPLICIT REAL*B(A-H.O-Z) CCMMON /ELKA/CTIME.DTIME.MM
	•	SPROB=1.DO K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR.I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 20*NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION THE ROUTINE ASSUMES THAT A SAMPLE OF SIZE SAMSIZ WILL BE GENERATED AND RETURNS ONLY ONE SAMFLE POINT PER CALL IMPLICIT REAL*8(A-H.O-Z)
		SPROB=1.DO K=JP DO 1 I=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR,I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB+(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB.CPTIM.CLTIM.SPROB.IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 20*NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION THE ROUTINE ASSUMES THAT A SAMPLE OF SIZE SAMSIZ WILL BE GENERATED AND RETURNS ONLY ONE SAMFLE POINT PER CALL IMPLICIT REAL*B(A-H.O-Z) CCMMON /ELKA/CTIME.DTIME.MM DIMENSION CTIME(0500).DTIME(0500)
		SPROB=1.00 K=JP DO 1 1=1.KJ IH=K/2 IZ=K-IH+2 L=INCLUS(IR,I+IJ) CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L) DTIME(L)=CTIME(L) SPROB=SPROB*(IZ*PQ(L)+(1-IZ)*PP(L)) K=IH RETURN END SUBROUTINE STRAT(NIB,CPTIM,CLTIM,SPROB,IY) STRAT SAMPLES SYSTEMATICALLY FROM THE 2**NINCL(IR) POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME AND PROBABILITY FOR EACH SAMPLED CONFIGURATION THE ROUTINE ASSUMES THAT A SAMPLE OF SIZE SAMSIZ WILL BE GENERATED AND RETURNS ONLY ONE SAMFLE POINT PER CALL IMPLICIT REAL*8(A-H,O-Z) CCMMON /ELKA/CTIME,DTIME,MM DIMENSION CTIME(0500),DTIME(0500)

٠.	CCMMON /BLKH/FHI, FLU, PP.PQ
	DIMENSION FHI (500) .FLO(500) .PP(500) .FQ(500)
	COMMON /BLKI/IEDF.NSUB.SAMSIZ
	INTEGER SAMSIZ
	COMMON /BLKK/IR.IB
	INTEGER+4 BSUM.BCGEF
	INTEGER+2 DIGIT(31)
	SPRUB=1.00
	IF(NIB.NE-1)GO TO 10
	INITIALIZE AND FIND THE FIRST CONFIGURATION
	TATTIALTZE AND FIND THE FIRST CONFIGURATION
	M=SAMS1Z
	N=NINCL (IR)
	NM=1
	BCOEF=1
	BSUM=BCOEF
	NP 1=N+1
	N1=N
	M 1 = M - 1
	S=OFLOAT(2**N-2)/OFLOAT(M-2)
	DC 5 II=1.N
	L=INCLUS(IR.II)
	SPROE=SPROB*PQ(L)
	DTIME(L)=FHI(L)
	CTIME(L)=FHI(L)
	CPTIM=CPTIME(CPATHT)
	CLTIM=CLTIME(CPATHT) CALL RANG(IY.X)
	RS=-5*X+2.D0
	GO TO 40
0	IF(NIB.GE.M)GU TO 30
	FIND THE NEXT CONFIGURATION IN THE SYSTEMATIC SAMPLE
	RS=RS+S
	IS=IDINT(RS)
	IF(BSUM.GE.IS)GO TO 2
	BCOEF=BCOEF*N1/(NP1-N1)
	N1=N1-1
	BSUM=BSUM+BCDEF
	60 10 1
	I=IS-BSUM+BCOEF CALL CONFIG(1.BCOEF.NI.N.DIGIT)
	CALL CONFIG(I.BCUEF.NI.N.DIGIT)
	COMPUTE CPTIM.CLTIM.AND SPROB
	COMPOSE CHIEFCESTRAND SPRUD
	DC 20 II=1.N
	L=INCLUS(IR.II)
	IF(DIGIT(II))18.18.19
	CTIME(L)=FLO(L)
3	
8	
8	DTIME(L)=FLO(L) SPRCB=SPRGB+PP(L)
8	DTIME(L)=FLO(L) SPRCB=SPRGB+PP(L)
,	DTIME(L)=FLO(L) SPRCB=SPRGB+PP(L)

```
SPROB=SPROB*PO(L)
                                                                                1525
20
      CONTINUE
                                                                                 1526
      CPTIM=CPTIME(CPATHT)
                                                                                1527
      CLTIM=CLTIME(CPATHT)
                                                                                1528
      GO TO 40
                                                                                1529
C
                                                                                1530
C
      FIND THE LAST CONFIGURATION
                                                                                1531
C
                                                                                1532
30
      DO 6 II=1.N
                                                                                1533
      L=INCLUS(IR.II)
                                                                                1534
      SPROB=SPROB*PP(L)
                                                                                1535
      DTIME(L)=FLG(L)
                                                                                1536
6
      CTIME(L)=FLO(L)
                                                                                1537
      CPTIM=CPTIME(CPATHT)
                                                                                1538
      CLTIM=CLTIME(CPATHT)
                                                                                1539
40
      RETURN
                                                                                1540
      END
                                                                                1541
      SUBROUTINE CONFIG(I.BCOEF.NI.N.DIGIT)
                                                                                1542
C
                                                                                1543
C
         CONFIG FINDS THE I-TH LARGEST N-CIGIT BINARY NUMBER
                                                                                1544
         HAVING NI- ONES AND (N - NI)- ZEROES . BCOEF IS THE TOTAL
C
                                                                                1545
C
         NUMBER OF SUCH CONFIGURATIONS
                                                                                1546
C
                                                                                1547
      IMPLICIT INTEGER*4(A-Z)
                                                                                1548
                                                                                1549
        . . . . . . . . . . . . . . . .
      INTEGER#2 DIGIT(31)
                                                                                1550
                                                                                1551
    . . . . . . . . . . . . .
      DO 1 11=1.N
                                                                                1552
      DIGIT(11)=0
                                                                                1553
      NIP=NI
                                                                                1554
      NP=N
                                                                                1555
      R=I
                                                                                1556
      INDEX=N
                                                                                1557
      C=BCCEF
                                                                                1558
      IF(NIP.LT.1)GO TO 6
                                                                                1559
      C=C+NIP/NP
                                                                                1560
      NIP=NIP-1
                                                                                1561
      NP=NP-1
                                                                                1562
      RR=R-C
                                                                                1563
      IF(RR)4.4.5
                                                                                1564
      DIGIT(INDEX)=1
                                                                                1565
      INDEX=INDEX-1
                                                                                1566
      GO TO 3
                                                                                1567
      INDEX= INCEX-1
                                                                                1568
      R=RR
                                                                                1569
      C=C+(NP-N1P)/NP
                                                                                1570
      NP=NP-1
                                                                                1571
      RR=R-C
                                                                                1572
      1F (RR)4.4.5
                                                                                1573
      RETURN
                                                                                1574
      END
                                                                                1575
      SUBROUTINE SIMPLE(NIB, CPTIM, CLTIM, SPROB, IY)
                                                                                1576
      IMPLICIT REAL +8(A-H.O-Z)
                                                                                1577
C
                                                                                1578
         SIPPLE GENERATES (AT RANDOW) ONE OF THE 2**NINCL(IR)
C
                                                                                1579
C
         POSSIBLE CONFIGURATIONS AND FINDS ITS COMPLETION TIME AND PROB
                                                                                1580
C
                                                                                1561
C
      INITIALIZE
                                                                                1582
                                                                                1583
                                                                                1584
c .
        . . . . . . . . . . . . . . . . . . .
      COMMON /BLKE/INCLUS.NINCL.NCLUS
                                                                                1585
```

	DIMENSION NINCL(50).INCLUS(50,100)	1586
C		1587
	COMMCN /BLKI/IEDF.NSUB.SAMSIZ	1588
	INTEGER SAMSIZ	1589
-		
c		1590
	CGMMON /BLKK/IR.IB	1591
C		1592
	SPROB=1.D0	1593
	IF(NIB.NE.11GD TO 10	1594
	NCLUS=NINCL(IR)	1595
	NNC=NCLUS/31	1596
	NNCR=NCLUS-NNC+31	1597
10	IF(NAC.EG.O)GO TO 20	1598
	RNCL=DFLQAT(2**31-1)	1599
		1600
	KJ=31	
	DG 1 1=1.NNC	1601
	CALL RANC(IY.U)	1602
	JP=IDINT(U*RNCL)+1	1603
	IF(NIB.EQ.1)JP=IDINT(RNCL+.5DO)	1604
	IF (NIB.EQ.SAMSIZ) JP=IDINT (RNCL+1.5D0)	1605
	그 아는 그는 그가 그 집에 가면서 하는 아는 아를 깨가 되었다. 이렇게 먹는 이 경험에 가장하는 생각이 되었다. 그는	
	[J=(I-1)*3]	1606
	CALL CONVRT (JP.IJ.KJ.SPROB)	1607
1	CONTINUE	1608
20	IF(NNCR.EQ.0)GD TO 25	1609
	RNCL=DFLGAT(2**NNCR-1)	1610
		1611
	CALL RANG(IY.U)	
	JP=IDINT(U*RNCL)+1	1612
	IF(NIB.EQ.1)JP=IDINT(RNCL+.5D0)	1613
	IF(NIB.EQ.SAMSIZ)JP=IDINT(RNCL+1.5D0)	1614
	IJ=NNC+31	1615
	CALL CONVRT(JP.IJ.NNCR.SPROB)	1616
25	CPTIM=CPTIME(CPATHT)	1617
	CLTIM=CLTIME(CPATHT)	1618
	RETURN	1619
	END .	1620
	SUBFOUTINE RAND(NSEED.U)	1621
c	ROUTINE FOR GENERATING UNIFORM RANDOM NUMBERS BETWEEN O AND	1622
C	1. BASED UPON A 32 BIT LENGTH INTEGER	1623
C		1624
C		1625
C	NSEED IS THE STARTING INTEGER CHOSEN FROM A RANDOM NUMBER TABLE	1626
c	OF TEN DIGIT INTEGERS	1627
	or ien bigit integers	
C		1628
C	U IS THE RETURNING UNIFORM RANDOM NUMBER	1629
C		1630
C		1631
c	MD IS THE LARGEST SINGLE PRECISION INTEGER (MD = 2147483647)	1632
-	FMD = FLCAT (MD)	1633
c	FFG - FECAL (NO)	
c		1634
c	SEE HOAGLIN FOR TEST OF RANDOMNESS	1635
•		1636
	IMPLICIT REAL+B(A-H.C-Z)	1637
	DATA FMD/2147483647.DU/.MD/2147483647/.ML/764261123/	1638
	ASEED=ASEED+ML	1639
	IF (MSEED-LT.O)MSEED=NSEED+mJ+1	1640
	U=DFLGAT(NSEED)/FMO	1641
	SE TURN	1642
	FAC	1643
	SUBMOUTINE SIMPLE	1644
		1645
2 2 2		
	LEASEN /BLX:/!EDF.NSUB.SAMSIZ	1646

```
INTEGER SAMSIZ
                                                                      1647
                                                                       1648
     COMMON /BLKY/FDUHAT.FOLHAT.FHAT
                                                                       1649
     DIMENSION FOLHAT (3,20), FOUHAT (3,20), FFAT (20)
                                                                       1650
     REAL *8 FCLHAT. FDUHAT. FHAT
                                                                       1651
1652
                                                                       1653
     REAL +4 THETA(3)
                                                                       1654
     1655
     COMMON /BLK1/81 INV. OMEGA.MM.KK.MMKK.MMKK2.N. PP1. JFIRST
                                                                       1656
                                                                       1657
     DIMENSION BRHS1(230).INBASE(230).XB1(230).Y1(230).B11NV(230.230)
                                                                      1658
     DIMENSION OMEGA(3)
                                                                       1659
     INTEGER Z
                                                                       1660
1661
     wT(Ow) = (OW+2.-3.)+Ow+Ow+1.
                                                                       1662
C
                                                                      1663
        THE FOLLOWING "TOLERANCES" ARE USED IN THE ALGORITHM.
C
                                                                       1664
        THEY MUST BE NON-NEGATIVE AND WOULD BE ZERO EXCEPT FOR THE
C
                                                                       1665
C
        NUMERICAL INACCURACY OF THE COMPUTER
                                                                      1666
             TGLRI : IF THE MAX REDUCED COST IS LESS THAN OR EQUAL TO
C
                                                                       1667
                    TOLRI THEN ALL REDUCED COSTS ARE CONSIDERED TO BE
c
                                                                      1668
C
                    NON-POSITIVE.
                                                                      1669
             TOLR2 : ANY COMPONENT Y(I.J) LESS THAN OR EQUAL TO TCLR2
C
                                                                       1670
C
                    IS CONSIDERED NON-POSITIVE.
                                                                       1671
C
                                                                       1672
     TOLR1=1.E-4
                                                                      1673
     TOLR2=1.E-4
                                                                      1674
     CALL ERRSET(208-256-1.1)
                                                                      1675
C
                                                                       1676
C
        THE INPUT
                                                                       1677
C
                                                                       1678
C
        M = THE NUMBER OF CONSTRAINTS NOT INCLUDING THE OBJECTIVE
                                                                       1679
c
           FUNCTION
                                                                      1680
C
        N = THE NUMBER OF VARIABLES
                                                                       1681
C
                                                                      1682
     JF IRST=0
                                                                      1683
     KK=3
                                                                      1684
                                                                      1685
     MM= LEDF-1
     M=4+MM+KK+1
                                                                       1686
     N=(6*KK+7)*MM+1
                                                                      1687
     MMKK=MM*KK
                                                                       1688
     MPKK2=MMKK#2
                                                                       1689
     MP 1=M+1
                                                                       1690
     DG 100 L=1.KK
                                                                       1691
100
     OMEGA(L)=1./(1.+THETA(L))
                                                                      1692
                                                                       1693
C
C
        BRHS IS A COLUMN OF CONSTANTS. THE I-TH ELEMENT
                                                                      1694
C
        OF BRHS IS THE CONSTANT ON THE RIGHT-HAND SIDE OF THE I-TH
                                                                       1695
C
        EQUALITY IN AX = BRHS
                                                                      1696
C
                                                                      1697
     BRHS1(1)=0.
                                                                       1698
                                                                       1699
     11=1
     DC 610 L=1.MM
                                                                       1700
     DO 610 I=1.KK
                                                                      1701
     11=11+1
                                                                      1702
     12=MMKK+I1
                                                                      1703
     13=MPKK+12
                                                                      1704
      14=PMKK+13
                                                                      1705
     BRHS1(11)=FOUHAT(1.L)
                                                                      1706
     BRHS1(12)=FOLHAT(1.L)
                                                                      1707
```

```
ERHS1(13)=BRHS1(11)
                                                                                 1708
610
      BRHS1(14)=BRHS1(12)
                                                                                 1709
      BRHS1 (MP1)=1.
                                                                                 1710
c
                                                                                 1711
c
         THE MPI-BY-N MATRIX AI IS THE AUGMENTED MATRIX OF CONSTANTS IN
                                                                                 1712
c
         THE CONSTRAINTS AX = BRHS. IT WILL BE GENERATED COLUMN-BY-
                                                                                 1713
c
         COLUMN AS NEEDED.
                                                                                 1714
c
                                                                                 1715
c
                                                                                 1716
C
         INBASE IS A SET OF M INTEGER VARIABLES WHICH INDICATE THE
                                                                                 1717
c
         COMPOSITION OF THE CURRENT BASIS B. FOR EXAMPLE.
                                                                                 1718
c
               INBASE(K) = 7 IMPLIES THAT THE K-TH COLUMN IN THE BASIS B
                                                                                 1719
C
                              CORRESPONDS TO THE 7-TH VARIABLE
                                                                                 1720
c
         INBASE(1) = 0 IMPLIES THAT THE FIRST COLUMN IN THE BASIS 82
                                                                                 1721
c
               CORRESPONDS TO XO
                                                                                 1722
C
                                                                                 1723
      INBASE(1)=0
                                                                                 1724
      IU=MMKK2+1
                                                                                 1725
      DO 21 K=2.1U
                                                                                 1726
      INEASE(K)=K-1
                                                                                 1727
      14=14+1
                                                                                 1728
      IL=MPKK2+7*MM-1
                                                                                 1729
      DO 52 K=1U.MP1
                                                                                 1730
52
      INBASE(K)=IL+K
                                                                                 1731
c
                                                                                 1732
C
         THE M+1 BY M+1 MATRIX BIINV IS THE INVERSE OF THE REVISED
                                                                                 1733
C
         SIMPLEX BASIS MATRIX B1.
                                                                                 1734
         THE FOLLOWING STATEMENTS CONSTRUCT THE INITIAL MATRIX BILINY
C
                                                                                 1735
C
                                                                                 1736
      DO 10 11=1.MP1
                                                                                 1737
      DO 12 L=1.MP1
                                                                                 1738
12
      B11NV(L.11)=0.
                                                                                 1739
      BIINV(II.II)=1.
10
                                                                                 1740
      IU=MMKK2+1
                                                                                 1741
      DO 11 II=2.IU
                                                                                 1742
      B1[NV(MMKK2+11.11)=1.
                                                                                 1743
      11=11-1
                                                                                 1744
      12=11/KK
                                                                                 1745
      L=11-12*KK
                                                                                 1746
      IF (L.EQ. 0) L=KK
                                                                                 1747
11
      BIINV(1.II) =-WT (DMEGA(L))
                                                                                 1748
C
                                                                                 1749
C
         CALCULATE THE INITIAL VALUES OF THE BASIC VARIABLES
                                                                                 1750
c
                                                                                 1751
      DO 753 I=1.MP1
                                                                                 1752
      XB1(1)=0.
                                                                                 1753
      DC 755 K=1.MP1
                                                                                 1754
755
      X81(1)=X81(1)+811NV(1.K)+BRHS1(K)
                                                                                 1755
753
      CONTINUE
                                                                                 1756
      CCATINUE
350
                                                                                 1757
                                                                                 1758
c
         START THE BASIC SIMPLEX ALGORITHM
                                                                                 1759
C
                                                                                 1760
C
         COMPUTE THE REDUCED COSTS: C(J) -Z(J)
                                                                                 1761
C
         THE J-TH REDUCED COST IS DENGTED BY REDCOS(J)
                                                                                 1762
C
                                                                                 1763
         FINC THE MAXIMUM REDUCED COST
c
                                                                                 1764
C
               RMAX = THE MAXIMUM REDUCED COST
                                                                                 1765
               IRMAX = THE SMALLEST INDEX J SUCH THAT REDCOS(J)=RMAX
                                                                                 1766
C
C
                                                                                 1767
      Z=1
                                                                                 1768
```

```
IRMAX=1
                                                                               1765
      RMAX=REDCST(Z.1)
                                                                               1770
      DC 24 J=2.N
                                                                               1771
      REDCOS=REDCSI(Z.J)
                                                                               1772
      IF (REDCOS.LE.RMAX)GO TO 24
                                                                               1773
      RMAX=REDCOS
                                                                               1774
      IRMAX=J
                                                                               1775
      CCATINUE
24
                                                                               1776
                                                                               1777
         IF RMAX IS NOT POSITIVE THEN ALL THE REDUCED COSTS ARE
C
                                                                               1778
c
         NON-POSITIVE AND THE CURRENT SOLUTION IS AN OPTIMAL SOLUTION
                                                                               1779
      IF(RMAX .LE. TOLRI) GO TJ 401
C
                                                                               1781
C
         DETERMINE THE IRMAX-TH YI VECTOR
                                                                               1782
C
                                                                               1783
      CALL YRMAX(IRMAX.MP1)
                                                                               1784
C
                                                                               1785
         IF ALL THE COMPONENTS OF Y ARE NON-POSITIVE. THE PROBLEM HAS
C
                                                                               1786
C
         AN UNBOUNDED SOLUTION. FOR THE PROBLEM BEING SOLVED THERE IS NO
                                                                               1787
C
         NEED TO CHECK FOR UNBOUNDEDNESS SINCE CX IS BOUNDED ABOVE BY 0.
                                                                               1788
C
                                                                               1789
         THE IRMAX-TH VARIABLE IS TO ENTER THE BASIS.
                                                                               1790
C
                                                                               1791
C
         THE VARIABLE TO LEAVE THE BASIS IS NOW DETERMINED.
                                                                               1792
              RMIN = THE MINIMUM RATIO OF XB(I)/Y(I) WITH Y(I) > 0.
C
                                                                               1793
              IRMIN = THE INDEX OF THE BASIS COLUMN TO BE REMOVED
                                                                               1794
                                                                               1795
      RMIN= . 99E20
                                                                               1796
      IRMIN=0
                                                                               1797
      DO 32 11=2.MP1
                                                                               1798
      IF(YI(II).LE. TOLR2) GO TO 32
                                                                               1799
      R=X81(11)/Y1(11)
                                                                               1800
      RR=R-RMIN
                                                                               1801
      IF (RR.GE.O.) GO TO 32
                                                                               1802
      GMIN=R
                                                                               1803
      IRMIN=11
                                                                               1804
32
      CONTINUE
                                                                               1805
                                                                               1806
         REPLACE THE IRMIN-TH COLUMN OF THE BASIS BY THE IRMAX-TH
                                                                               1807
C
         COLUMN OF A
                                                                               1808
C
C
         UPDATE THE BASIS INVERSE: BLINV
                                                                               1810
                                                                               1811
      DO 33 J=2.MP1
                                                                               1812
      WW=B1INV(IRMIN .J)/Y1(IRMIN )
                                                                               1813
      DO 37 L=1.MP1
                                                                               1814
                                                                               1815
37
      BIINV(L.J)=BIINV(L.J)-W##Y1(L)
                                                                               1816
33
      BIINV(IRMIN .J)=WW
                                                                               1817
C
         UPDATE THE BASIC VARIABLES: INBASE AND XB1
C
                                                                               1819
      INBASE (IRMIN) = IKMAX
                                                                               1820
      W=XB1(IRMIN )/Y1(IRMIN )
                                                                               1821
      DQ 38 1=1.MP1
                                                                               1822
      XB1(1)=XB1(1)-Y1(1)+W
                                                                               1823
      XBI(IRMIN )=W
                                                                               1824
      GO TO 350
                                                                               1825
401
       CCNTINUE
                                                                               1826
      CONTINUE
                                                                               1827
999
      DO 1010 L=1.MM
                                                                               1828
1010 FHAT(L)=0.00
                                                                               1829
```

	DC 1011 L=1,MP1	
	KKK=INBASE(L) IF(KKK.GT.MMKK2.AND.KKK.LE.MMKK2+MM) FHAT	(KKK-MMKK2)=XB1(L)
1011	CONTINUE	TRANSPORTE /
	DO 1013 L=2.MM	
1013	FHAT(L)=FHAT(L)+FHAT(L-1)	
••••	FHAT (IEDF)=1.00	
	RETURN	
	END	
	SUBROUTINE YRMAX(IRMAX.MP1)	
c		
	COMMON /BLK2/Y1	
	DIMENSION Y1(230)	
	INTEGER Z	
c		
	00 1 Z=1.MP1	
1	Y1(Z)=-REDCST(Z.IRMAX)	
	RETURN	
	END	
	FUNCTION REDCST(Z.1)	
	IMPLICIT INTEGER#4 (C-Z)	
	COMMON /BLK1/B1INV.W.M.K.MK.MK2.NN.NP1.JF	FIRST
	REAL*4 BIINV(230,230)	
	REAL *4 W(3), WWT(3), WW(3,3), WT, CW, REDCST, C	OR .
	WT(UW) = (UW * 2 3.) * UW * UW * 1.	
	IF(JFIRST.NE.O)GG TO 101	
	JFIRST=1	
	LL1=MK2	
	LL2=LL1+M	
	LL3=LL2+6*M	
	LL4=NN LQ=MK2+1	
	DC 100 J=1.K	
	Ow=W(J)	
	WW(J.1)=GW	
	ww(J.2)=Cw+Ow	
	WW(J.3)=-OW+OW	
00	hhT(J)=hT(Om)	
101	IF(1.GT.LL1)GO TO 200	
	P=I+1	
	REDCST=-B1[NV(Z.P)+B1[NV(Z.MK2+P)	
	IF(Z.NE.1)GO TO 500	
	P=1/K	
	Q=I-K*P	
	1F(Q.EQ.0)Q=K	
	REDCST=REDCST-WWT(Q)	
	GO TO 500	
00	IF(1.GT.LL2)GO TO 300	
	REDCST=0.	
	0=1-MK2	
	1L=-WK+(Q-1)*K+2	
	IU=1	
	DC 202 P=1.4	
	IL=IL+MK	
	IU=IU+MK	
	DC 202 L=IL.IU	
202	REDCST=REDCST-B1INV(Z.L)	
	REDCST=REDCST-BIINV(Z.NPI)	
	GC TO 500	

300	IF(I.GT.LL3)GO TO 400 REDCST=0.	1891
	L=I-LL2	1893
	P=L/3	1894
	Q=L-3*P	1895
	IF(C.NE.0)GO TO 303	1896
	Q=3	1897
	P=P-1	1898
303	IL=P*K+1	1899
	IU=MK2+IL	1900
	DC 302 R=1.K	1901
	DR=WW(R,G)	1902
	IF (P.GE.M.AND.Q.EQ.1)DR=~DR	1903
302	REDCST=REDCST-OR*(B1 INV(Z.IL+R)+B1 INV(Z.IU+R))	1904
	GC TO 500	1905
400	IF(1.GT.LL4)GD TO 500	1906
	Q=I-LL3+1	1907
	IF(G-LG)401.401.402	1908
401	REDCST=B1INV(Z,Q)	1909
	GO TC 500	1910
402	REDCSI=-B11NV(Z.Q)	1911
500	RETURN	1912
300		
	END	1913

THE RESIDENCE OF THE THE INSTRUCTION OF THE THEORY OF THE THE PROPERTY OF THE THEORY OF THEORY OF THE THEORY OF THEORY OF THE TH

```
8.
   SYNTH
      SYNTHESIS PROGRAM
   SYNTHESIS COMBINES PARALLEL AND SERIES SUBNETWORKS IDENTIFIED BY
C
   THE DECCMPGSITIUN PROGRAM INCORPORATING SUBNETWORK DURATION
   DISTRIBUTION INFORMATION PROVIDED BY THE SUBNETWORK ANALYSIS PROGRAM
C
   THE FULLCWING IS AN ALPHABETICAL LIST OF THE VARIABLES AND ARRAYS
   USED BY BOTH THE MAIN PROGRAM AND ITS SUBROUTINES
C FX(1.J)=JTH C.D.F. VALUE FOR THE ITH SURNETWORK
                                                                                 10
   FXSAVE(1)=ITH C.D.F. VALUE RESULTING FROM THE CURRENTLY SYNTHESIZED
C
                                                                                 11
   SLBNETWORKS
                                                                                 12
   K1=NUMBER OF THE 1ST SUBNETWORK IN PAIR CURRENTLY BEING SYNTHESIZED
C
                                                                                 13
   K2=NUMBER OF THE 2ND SUBNETWORK IN PAIR CURRENTLY BEING SYNTHESIZED
                                                                                 14
   NIEDF(I)=NUMBER OF PCINTS IN THE ITH SUBNETWORK DISTRIBUTION
                                                                                 15
   NT=NUMBER OF C.D.F. SUBDIVISIONS USED THROUGHOUT SYNTHESIS (MAX=99)
                                                                                 16
   NUMSUB = NUMBER OF SUBNETWORKS
                                                                                 17
   X(I.J)=JTH X VALUE FOR THE ITH SUBNETWORK
   XSAVE(1)=ITH X VALUE RESULTING FROM THE CURRENTLY SYNTHESIZED SUBNETW
                                                                                 20
   THE FULLEWING IS AN ALPHABETICAL LIST OF THE VARIABLES AND ARRAYS USE
                                                                                 21
C
   CNLY IN THE MAIN PROGRAM
                                                                                 22
                                                                                 23
   ID=NUMBER OF THE SUBNETWORK
   IEDF=NUMBER OF PUINTS IN THE C.D.F.
C
                                                                                 25
   INSNG(1)=ITH INSTRUCTION NUMBER
                                                                                 26
C
   IOPT=0 INDICATES S WILL BE CHOSEN AS A PERCENTILE OF THE SYNTHESIZED
                                                                                 27
          DISTRIBUTION
                                                                                 28
       =1 INCICATES S WILL BE THE MEAN OF THE SYNTHESIZED DISTRIBUTION
   ISORP(I)=0 INDICATES THE CURRENT SUBNETWORKS ARE IN SERIES IN THE
C
                                                                                 30
               ITH INSTRUCTION
                                                                                 31
C
           =1 INDICATES THE CURRENT SUBNETWORKS ARE IN PARALLEL IN THE
                                                                                 32
C
               ITH INSTRUCTION
                                                                                 33
   ISUBNT(1)=SUBNETWORK RESULTING FROM THE ITH INSTRUCTION
C
C
   JSUBNI(I.J)=JTH SUBNET WORK CONTAINED IN THE ITH INSTRUCTION
                                                                                 35
               =0 INDICATES THE PREVIOUS NUMBER WAS THE LAST SUBNETWORK F
-
                  THIS INSTRUCTION
                                                                                 37
   ACINS=NUMBER OF INSTRUCTIONS TO BE SYNTHESIZED
                                                                                 38
   PCT=PERCENTILE VALUE TO BE USED IN DETERMINING S
                                                                                 39
C
   PD=DESIRED PROJECT DEADLINE
                                                                                 40
   S=MEAN OR PERCENTILE OF FINAL DISTRIBUTION
                                                                                 41
   TT=TARGET TIME USED FOR THE TIME COMPRESSION PROGRAM
C
                                                                                 42
C
                                                                                 43
         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS. LET
C
                                                                                 44
C
         NTMAX = THE MAXIMUM NUMBER OF SUBDIVIDIONS ALLOWED IN THE COF
                                                                                 45
         NTMAX = THE MAXIMUM NUMBER OF SUBCIVISIONS ALLOWED IN THE COF
C
                                                                                 46
                     FOR THE ENTIRE PROJECT
C
                                                                                 47
         NSUEMAX = THE MAXIMUM NUMBER OF SUBNETOWRKS ALLOWED IN THE
C
                     DECOMPOSITION PROCESS
                                                                                 40
C
         IEDFMAX = THE MAXIMUM NUMBER OF SUBDIVISIONS ALLOWED IN THE
                                                                                 50
                      APPROXIMATE COF FOR EACH SUBNETWORK
C
                                                                                 51
         ACINSMAX = THE MAXIMUM NUMBER OF INSTRUCTIONS GENERATED BY THE
C
                                                                                 52
                      DECOMPOSITION PROCESS
                                                                                 53
         MAX = MAXIMUM OF (NTMAX. IEDFMAX)
                                                                                 54
C
         CURRENTLY. NTMAX=50: NSUBMAX=100: IEDFMAX=20: NOINSMAX=100
C
                                                                                 56
C
      DIMENSION X(NSUBMAX.MAX).FX(NSUBMAX.MAX).FXSAVE(NTMAX).
                                                                                 57
C
                 NIEDF (NSUBMAX) . INSNO(NCINSMAX) . ISORP(NCINSMAX) .
                                                                                 58
C
                 ISUBNI (NOINSMAX). JSUBNI (NOINSMAX. 24).R (NTMAX). S(NTMAX).
                                                                                 59
                 FR(NTMAX) .FS(NTMAX) .FILE5(NGINSMAX #27) .
                                                                                 60
```

```
EGUIVLANECE (FILES(NGINSMAX+1).ISUBNT(1)).
C
                  (FILES(2*NCINSMAX+1).ISORP(1)).
                                                                                  62
C
                  FILES(3*NOINSMAX+1).JSUBNT(1))
                                                                                  63
                                                                                  64
      IMPLICIT REAL+8 (A-H.O-Z)
                                                                                  65
      CCMMON /BLKA/X(100.50).FX(100.50).XSAVE(50).FXSAVE(50).
                                                                                  66
     *NIEDF(100).NI
                                                                                  67
      CCMMON /BLKB/R( 50).S( 50).FR( 50).FS( 50)
                                                                                  68
      REAL#4 TTT. DPD. DCT
                                                                                  69
      DIMENSION INSNO(100).ISORP(100).ISUBNT(100).JSUBNT(100.24)
                                                                                  70
      INTEGER#4 FILES(2700)
                                                                                  71
      INTEGER
                                         F4/12/.F5/13/
                                                                                  72
      EGUIVALENCE (FILES(1), INSNO(1)). (FILES(101), ISUBNT(1)),
                                                                                  73
                  (FILE5(201).ISORP(1)).(FILE5(301).JSUBNT(1))
                                                                                  75
   READ DATA IN AND PRINT INPUT INFORMATION
                                                                                  76
                                                                                  77
                                                                                  78
      MRITE (6.200)
200
     FGFMAT(1H1.132('*')/1x.132('*')/
                                                                                  79
     **OTHIS IS THE OUTPUT FROM THE SYNTHESIS PROGRAM: SYNTH*/
                                                                                  AO
     *1H0,132(***)/1x,132(***)/////* THE CURRENT PROJECT SCHEDULE IS TH
                                                                                  18
     *E ONE MOST RECENTLY LISTED BY THE DETERMINISTIC SCHEDULE RESCLUTIO
                                                                                  82
     *N PROGRAM.*/*OTHE PROJECT*,1H*,*S CORRESPONDING APPROXIMATE COMPLE
                                                                                  83
     *TIOM TIME DISTRIBUTION IS DETERMINED BELGW: 1)
      REMIND F4
                                                                                  85
      REWIND F5
                                                                                  86
      READ (F4 )NCYC.TTT.NFLAG
                                                                                  87
      READ(F5) IUPT.NT.DCT.DPD
                                                                                  88
      PCT = DCT
                                                                                  89
      TT = TTT
                                                                                  90
      PD = DPD
                                                                                  91
      ATP1=AT+1
                                                                                  92
      WRITE(6.11)NT.TT
                                                                                  93
      FORMAT(1HO// FOR THIS PROBLEM */ OTHE NUMBER OF C.D.F. SUBDIVISION
                                                                                  94
     +S USED THROUGHOUT SYNTHESIS IS .. . . . . THE TARGET TIME USED BY T
                                                                                  95
     *HE DETERMINISTIC SCHEDULER WAS .. F10.5)
                                                                                  96
      READ(FS) NOINS
                                                                                  97
      IF(ICPT.EQ.0) GO TO 17
                                                                                  98
      BRITE(6.16)PD
                                                                                  99
      FORMAT( THE MEAN OF THE PROJECT . 14 . S APPROXIMATE COMPLETION TI
                                                                                 100
     *ME DISTRIBUTION WILL BE COMPARED TO THE 1/0 SPECIFIED PROJECT DEADL
                                                                                 101
     *INE TIME OF ..................
                                                                                 102
      GC TO 19
                                                                                 103
   17 P=PCT+100.
                                                                                 104
      BRITE(6.18) P.PD
                                                                                 105
      FGRMAT( THE '.F6.2, '-TH PERCENTILE OF THE PROJECT', IH'. 'S APPROXI
                                                                                 106
     *PATE COMPLETION TIME DISTRIBUTION WILL BE COMPARED TO THE !/
                                                                                 107
     * * SPECIFIED PROJECT DEADLINE TIME OF '.F12.5)
                                                                                 108
   19 WRITE(6.600)
                                                                                 109
                    10x. THE INSTRUCTIONS TO BE PERFORMED ARE . . //)
  600 FCFMAT(//.
                                                                                 110
      WRITE(6.605)
                                                                                 111
  605 FCRMAT(T40, *SERIES=0*)
                                                                                 112
      WAITE (6.610)
                                                                                 113
  610 FCFMAT( 5x, INSTRUCTION NO. 1, 125, SUBNETWORK 1, 140, PARALLEL=11,
                                                                                 114
     $155. SUBNETWORKS TO BE SYNTHESIZED')
                                                                                 115
      READ(F5) FILES
                                                                                 116
      DO 20 1=1.NGINS
                                                                                 117
      NSP=0
                                                                                 118
                                                                                 119
      DC 210 J=1.24
      IF (JSUBNT(1.J).LE.0)GO TO 211
                                                                                 120
210
      ASP=NSP+1
                                                                                 121
```

```
CCNTINLE
211
      WRITE(6,26) INSNU(1), ISUBNT(1), ISORP(1), (JSUBNT(1,J), J=1,NSP)
                                                                                123
   26 FORMAT (/.T11.13.T28.13.T43.12.T55.2413)
                                                                                124
                                                                                125
   20 CONTINUE
      WRITE(6.31)
                                                                                126
   31 FCRMAT(1H1.//.10x. SUBNETWORK .T30. APPROXIMATE CISTRIBUTION DETER
                                                                                127
     SMINED USING SUBNETWORK ANALYSIS!)
                                                                                128
                                                                                129
35
      READ(F5.END=999) ID. IEDF.(X(ID.J).FX(ID.J).J=1.IEDF)
      NIEDF(ID)=IEDF
                                                                                130
      BRITE(6.51) 10
                                                                                131
   51 FGRMAT(/.T14.13.T34.'X'.T48.'F(X)')
                                                                                132
      WRITE(6.52) (X(ID.J).FX(ID.J).J=1.IEDF)
                                                                                133
   52 FCRMAT( T25.F15.5.T39.F15.5)
                                                                                134
      GO TO 35
C
                                                                                136
C
  CONSIDER INSTRUCTIONS IN REVERSE GROER AND SYNTHESIZE ALL SUBNETWORKS
                                                                                137
  ACCORDING TO SERIES OR PARALLEL
                                                                                138
C
                                                                                139
  999 DO 100 I=1.NUINS
                                                                                140
      II=NCINS-I+1
                                                                                141
      DO 80 J=1.24
                                                                                142
      KI=JSUBNT(II.J)
                                                                                143
      JP1=J+1
                                                                                140
      K2=JSUBNT(11.JP1)
                                                                                145
      IF(K2.EQ.0) GO TO 75
                                                                                146
      IF(ISORP(II).EQ.1) GO TO 55
                                                                                147
      CALL SERIES (KI.K2)
                                                                                148
      GO TO 60
                                                                                149
   55 CALL PARAL(KI.K2)
                                                                                150
C
                                                                                151
   STORE CURRENT DISTRIBUTION IN SUBNETWORK K2
                                                                                152
                                                                                153
   60 00 65 K=1.NTP1
                                                                                154
      X(K2.K)=XSAVE(K)
                                                                                155
      FX(K2.K)=FXSAVE(K)
                                                                                156
   65 CONTINUE
                                                                                157
      NIEDF (K2)=NTP1
                                                                                158
      GG TO 80
                                                                                159
C
                                                                                160
C
   STORE DISTRIBUTION RESULTING FROM CURRENT INSTRUCTION IN SUBNETWORK
                                                                                161
   GIVEN BY INSTRUCTION
                                                                                162
C
                                                                                163
   75 DO 78 K=1,NTP1
                                                                                164
      X(ISUBNT(II),K)=XSAVE(K)
                                                                               165
      FX(ISUENT(II),K)=FXSAVE(K)
                                                                                166
   78 CENTINUE
                                                                                167
      NIEDF (ISUENT(II))=NTPL
                                                                                168
      GC TO 100
                                                                                169
   80 CENTINUE
                                                                               170
  100 CUNTINUE
C
                                                                                172
  CUTPUT FINAL DISTRIBUTION
C
                                                                               173
                                                                               174
                                                                               175
      BRITE(6.105)
  105 FORMAT(1H1.//.10X. THE DISTRIBUTION FOR THE SYNTHESIZED NETWORK IS
                                                                                176
     $'.//.T15.'T'.T30.'F(T)')
                                                                                177
      DO 110 K=1.NTP1
                                                                                178
      IF (PCT.GT.FX(1.K)) KEEP=K
                                                                                179
  110 WRITE(6.120) X(1.K).FX(1.K)
                                                                                180
  120 FORMAT( 5x.2F15.5)
                                                                                181
                                                                                182
```

```
COMPUTE S AND PRODUCE INFORMATION NEEDED FOR TIME COMPRESSION
                                                                                184
      IF(ICPT.EG.O) GO TO 140
                                                                                185
      S=X(1.1)*FX(1.1)
                                                                                186
      DO 130 K=2.NTP1
                                                                                187
      KM1=K-1
                                                                                IRA
  130 S=S+X(1.K)*(FX(1.K)-FX(1.KM1))
                                                                                189
      GC TC 150
                                                                                190
  140 KEEP1=KEEP+1
                                                                                191
      DIFF1=X(1.KEEP1)-X(1.KEEP)
                                                                                192
      DIFF2=FX(1.KEEP1)-FX(1.KEEP)
                                                                                193
      DIFF3=PCT-FX(1.KEEP)
                                                                                194
      S=X(1.KEEP)+DIFF1+DIFF3/DIFF2
                                                                                195
  150 TT=PD+TT/S
                                                                                196
      ERRPCT=DABS(S-PD)+100./PD
                                                                                197
      IF(10PT)1.1.2
                                                                                198
1
      P=PCT+100.
                                                                                199
      BRITE(6.41P.S
                                                                                200
      FORMAT(//.10x. THE .. F6.2. -TH PERCENTILE OF THE SYNTHESIZED DISTR
                                                                                201
     * IBUTION IS '.F12.5./)
                                                                                202
                                                                                203
2
      BRITE (6.5)S
                                                                                204
      FORMAT (//10x THE MEAN OF THE SYNTHESIZED DISTRIBUTION IS .. F12.5/)
                                                                                205
3
      WALTE (6.180) S. ERRPCT.TT
                                                                                206
      FORMAT (
                10x. THE DIFFERENCE BETWEEN "F12.5." AND THE PROJECT DEA
                                                                                207
     *DLINE IS ",F9.2," PERCENT OF THE PROJECT DEADLINE."/
                                                                                208
     $10x, HENCE, THE NEW TARGET TIME IS . F12.5)
                                                                                209
      WRITE(6.300 INCYC
                                                                                210
      FORMAT(//1H0.132(***)/*OTHIS COMPLETES ITERATION*.17/1H0.132(***))
300
                                                                                211
      NCYC=NCYC+1
                                                                                212
      TIT=II
                                                                                213
      REWIND F4
                                                                                214
      WRITE (F4 INCYC.TTT.NFLAG
                                                                                215
  190 STOP
                                                                                216
      ENC
                                                                                217
C
                                                                                218
 THE FOLLOWING IS AN ALPHABETICAL LIST OF THE VARIABLES AND ARRAYS USE
                                                                                219
C
 IN BOTH SUBROUTINES
                                                                                220
                                                                                221
  BT=BEGINNING TIME FOR RESULTING DISTRIBUTION
                                                                                222
  DELT=INCREMENT FOR THE X VALUES ALLOWING FOR NT EQUAL DIVISIONS
C
                                                                                223
  FR(1)=1TH C.D.F. VALUE FUR THE 1ST SUBNETWORK OF PAIR TO BE SYNTHESIZ
                                                                                224
  FS(1)=ITH C.D.F. VALUE FOR THE 2ND SUBNETWORK OF PAIR TO BE SYNTHESIZ
                                                                                225
  NR=NUMBER OF POINTS IN 1ST SUBNETWORK
                                                                                226
   NS=NUMBER OF POINTS IN 2ND SUBNETWORK
                                                                                227
  R(1)=ITH X VALUE FOR THE 1ST SUBNETWORK OF PAIR TO BE SYNTHESIZED
                                                                                228
  S(I)=ITH X VALUE FOR THE 2ND SUBNETWORK OF PAIR TO BE SYNTHESIZED
                                                                                229
  T=X VALUE FOR THE RESULTING SYNTHESIZED SUBNETWORK
                                                                                230
  XMAX=MAXIMUM POSSIBLE X VALUE FOR THE RESULTING SYNTHESIZED SUBNETWCR
                                                                                231
   XWIN=MINIMUM POSSIBLE X VALUE FOR THE RESULTING SYNTHESIZED SUBNETWOR
C
                                                                                232
                                                                                233
      SUBROUTINE PARAL(K1.K2)
                                                                                234
                                                                                235
   SYNTHESIZES 2 PARALLEL SUBNETWORKS
                                                                                236
                                                                                237
      IMPLICIT REAL+8 (A-H.O-Z)
                                                                                238
      COMMON /BLKA/X(100.50),FX(100.50),XSAVE(50).FXSAVE(50).
                                                                                239
     *NIEDF (100) . NT
                                                                                240
      CCPMGN /BLKB/H( 50).S( 50).FR( 50).FS( 50)
                                                                                241
      NR=NIEDF(KI)
                                                                                242
      NS=NIEDF(K2)
                                                                                243
```

	DC 10 I=1,NH R(I)=X(K1,I)	24
10	FR(1)=FX(K1,1)	24
	DO 15 I=1.NS	24
	S(1)=x(K2,1)	24
15	F5(1)=FX(K2,1)	24
• •	XMIN=DMIN1(R(1).S(1))	25
	XMAX=CMAX1(R(NR).S(NS))	25
	BT=XMIN	25
	DELT=(XMAX-XMIN)/NT	25
	T=8T-DELT	25
	DC 50 I=1.NT	25
	F1=0.0D0	25
	F2=0.000	25
	T=T+DELT	25
	DO 40 J=1.NR	25
	IF(R(J).GT.T) GO TO 41	26
	F1=FR(J)	26
40	CONTINUE	26
	DC 45 J=1.NS	26
٠.	1F(S(J).GT.T) GD TO 46	26
	F2=F5(J)	26
46	CONTINUE	26
		26
40	FXSAVE(1)=F1*F2	26
= -	XSAVE(I)=T	
50	CONTINUE	26
	NTP1=NT+1	27
	FXSAVE(NTP1)=1.0D0	27
	XSAVE(NTP1)=XMAX	27
	RETURN	27
	END	27
	SUBROUTINE SERIES(KI.K2)	27
	THESIZES A SUBJETHERING IN CERTIC	27
311	THESIZES 2 SUBNETWORKS IN SERIES	27
	IMPLICIT REAL+8 (A-H.O-Z)	27
	CCMMGN /BLKA/X(100.50).FX(100.50).XSAVE(50).FXSAVE(50).	28
	NIEDF (100) NT	28
	CCMMON /BLKH/R(50).S(50).FR(50).FS(50)	28
	NR=NIECF(KI)	26
	NS=N1EDF (K2)	28
	DO 10 1=1.NR	21
	R(1)=x(K1,1)	2
	FR(1)=FX(K1.1)	2
••	DO 15 1=1.NS	21
	S(1)=x(K2,1)	21
15	FS(1)=FX(K2,1)	29
	XMIN=R(1)+S(1)	29
	XMAX=R(NR)+S(NS)	29
	BI=XMIN	29
	DELT=(XMAX-XMIN)/NT	
	T=8T-DELT	29
		29
	00 55 [=1.NT	29
	F1=0.000	29
	T=T+DELT	29
	DO 50 J=1.NS	5
	IF(S(J).GT.T) GO TO 54	30
	IF(J.6T.1) 60 TO 30	30
	FF=FS(1)	30
	GC TC 35	30
30	JJ=J-1	30

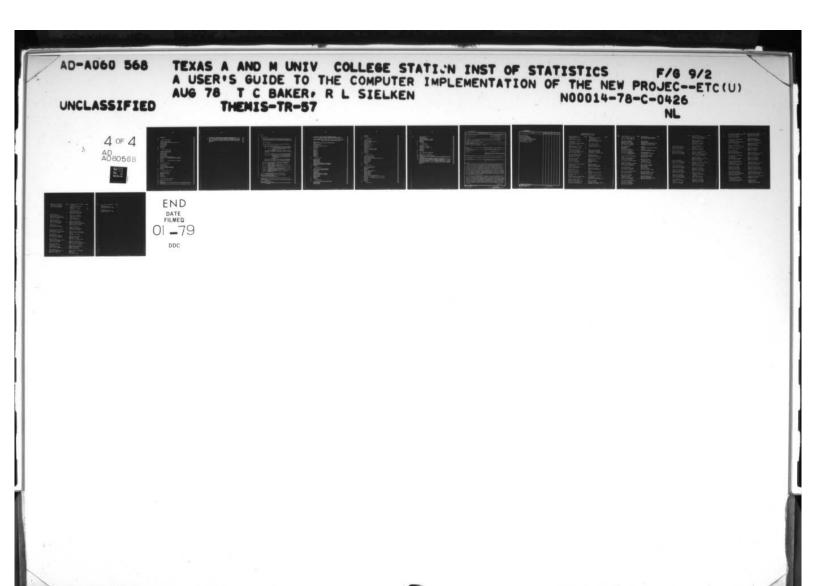
	FF=FS(J)-FS(JJ)	305
35	FTMS=0.000	306
	TMS=T-S(J)	307
	00 40 K=1,NR	308
	IF(R(K).GT.TMS) GO TO 45	309
	FTMS=FR(K)	310
40	CONTINUE	311
45	FT=FT+FF*FTMS	312
50	CGNTINUE	313
54	XSAVE(1)=T	314
	FXSAVE(1)=FT	315
55	CONTINUE	316
	NTP1=NT+1	317
	FXSAVE(NTP1)=1.000	318
	XSAVE(NTP1)=XMAX	319
	RETURN	320
	END	321

1

9. SAVEFIL

```
C
      PROGRAM SAVEFIL
C
                                                                                   2
c
         THIS PROGRAM TRANSFERS THE OPERATIONAL INFOPMATION GENERATED
                                                                                    3
C
         BY THE STATISTICAL PERT PROGRAMS FROM THE TEMPORARY FILES TO
C
         THE PERMANENT DATA SET DEFINED AS I/O-UNIT PS
C
C
C
         INPUT INSTRUCTIONS
C
C
         A SINGLE CARD SHOULD BE SUPPLIED AS INPUT TO THIS PROGRAM.
                                                                                   10
C
         THE CARD SHOULD BEAR IN COLUMNS 1 - B THE NAME OF THE LAST
c
         PROGRAM EXECUTED BEFORE THIS INTERRUPT.
                                                                                  12
c
         (FCRMAT 2A4. LEFT JUSTIFIED)
                                                                                   13
C
                                                                                  14
         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS. LET
C
                                                                                  15
C
         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL
c
                      PROJECT NETWORK
                                                                                  17
C
         BKMAX = THE MAXIMUM NUMBER OF BREAK POINTS ALLOWED IN THE
                      ENTIRE PROJECT'S TIME-COST CURVE
C
                                                                                   15
c
         NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR
                                                                                  20
c
                     EACH ACTIVITY
C
         IEDFMAX = THE MAXIMUM NUMBER OF SUBDIVISIONS ALLOWED IN THE
                                                                                  22
C
                      APPROXIMATE COF FOR EACH SUBNETWORK
                                                                                  23
         NCINSMAX = THE MAXIMUM NUMBER OF INSTRUCTIONS GENERATED BY THE
C
                                                                                  24
C
                     DECOMPOSITION PROCESS
                                                                                  25
C
                                                                                  26
C
      INTEGER*4 FILEO(4*MMAX+5).NCT(MMAX).FILE38(3*MMAX+5).
                                                                                  27
C
                FILES(NOINSMAX#27)
                                                                                   28
      INTEGER + 2 BREAK (BKMAX) . TIMCST (2 + NCTMAX) . FILE6(2 + MMAX)
c
                                                                                  29
      REAL+8 FD(IEDFMAX) . X(IEDFMAX)
C
                                                                                  30
c
      EQUIVALENCE (FILEO(3*MMAX+6).NCT(1))
                                                                                  31
C
                                                                                  32
                                                                                  33
      INTEGER F0/8/.F1/9/.F2/10/.F3/11/.F4/12/.F5/13/.F6/14/.F7/15/
                                                                                  34
      INTEGER PS/4/
      INTEGER FILEO(4005).NCT(1000).FILE3A(11).FILE3B(3005).FILE5(2700).
                                                                                  36
              LIST(24)
                                                                                  37
      REAL FILE1(5),FILE2(5)
                                                                                  38
      REAL+8 COF(12).FD(20).X(20)
                                                                                  39
      INTEGER*2 NBREAK.BREAK(3000).TIMCST(12).TEST1.TEST2.TEST3.STIME.
                FILE6(2000)
                                                                                  41
      EQUIVALENCE (FILEO(1).NACT). (FILEO(3006).NCT(1))
                                                                                  42
      EQUIVALENCE (FILE1(1).FILE2(1).FILE3A(1).FILE3B(1).FILE5(1).
                                                                                  43
                  FILE6(1), BREAK(1), TIMCST(1), LIST(1), CDF(1), FD(1))
                                                                                   44
      INTEGER IPROGIZI
                                                                                   45
      INTEGER PROG(13)/'MAIN'. 'LOOP'. 'DPS '. 'DSR '. 'SIMP'. 'MODS'. 'DECO'.
                                                                                  46
                        "SUBN", "SYNT", "DSR ", "MCDS", "SUBN", "SYNT"/
                                                                                  47
C
                                                                                   48
      WRITE(6.112)
      REBING FA
                                                                                  50
      READ (F4) NCYC
                                                                                  51
      READ(5,110) IPROG
                                                                                  52
                                                                                  53
      1F (NCYC.GE.2)K1=10
                                                                                  54
      CC 1 1=K1.13
                                                                                  55
      IF (IPRCG(1).EQ.PROG(1))GC TO 2
                                                                                  56
      CENTINUE
                                                                                  57
      BRITE(6,111)
                                                                                  58
      GO TO 109
                                                                                  59
      K=I
                                                                                  60
```

```
REWIND FO
                                                                                      61
      REBIND F1
                                                                                      62
      REMIND F2
                                                                                      63
      REWING F3
                                                                                      64
      REWING F4
                                                                                      65
      REWIND F5
                                                                                      66
      REBING F6
                                                                                      67
      REWIND F7
                                                                                      68
      REWING PS
                                                                                      69
C
                                                                                      70
         READ AND STORE FILEO
C
                                                                                      71
C
                                                                                      72
      READ(FO) FILEO
                                                                                      73
      WRITE (PS)FILEO
                                                                                      74
      READ(FO) JMAT
                                                                                      75
      WRITE(PS)JMAT
                                                                                      76
      IF(K.LT.3)GO TO 3
                                                                                      77
      READ(FO) NBREAK. (BREAK(I). [=1.NBREAK)
                                                                                      78
      BFITE (PS) NBREAK, (BREAK(I), I=1. NBREAK)
                                                                                      79
3
      CCATINUE
                                                                                      80
                                                                                      81
C
         REAC AND STORE FILE!
                                                                                      82
C
                                                                                      83
      READ(F1) TEST1. TEST2. TEST3. STIME
                                                                                      84
      WALTE(PS) TEST1, TEST2, TEST3, STIME
                                                                                      85
      DC 4 I=1. NACT
                                                                                      86
      NNCT=NCT(1)
                                                                                      87
      ANCTI=NNCT-1
                                                                                      88
      NNCT2=2*NNCT
                                                                                      89
      READ(F1) (TIMCST(J).J=1.NACT2)
                                                                                      90
      WRITE(PS)(TIMCST(J).J=1.NNCT2)
                                                                                      91
      DO 4 J=1 . NNCT1
                                                                                      92
      REAC(F1) FILE1
                                                                                      93
      BRITE(PS)FILEL
                                                                                      94
      CONTINUE .
                                                                                      95
c
                                                                                      96
C
          READ AND STORE FILE2
                                                                                      97
C
                                                                                      98
      IF(K.NE.4.AND.K.NE.5.AND.K.NE.10)GO TO 6
                                                                                      99
      DO 5 1=1.NACT
                                                                                     100
      REAC(F2) FILE2
                                                                                     101
      WRITE (PS)FILE2
                                                                                     102
5
      CENTINUE
                                                                                     103
      CCNTINUE
6
                                                                                     104
                                                                                     105
C
C
          READ AND STORE FILES
                                                                                     106
C
                                                                                     107
      READ(F3) FILE3A
                                                                                     108
      WRITE(PS)FILE3A
                                                                                    109
      IF (K.NE. 5. AND . K. NE. 61GO TO 7
                                                                                     110
      READ(F3) FILE38
                                                                                     111
      WALTE (PS)FILE38
                                                                                     112
7
      CENTINUE
                                                                                     113
      IF(K.LT.7)G0 TO 10
                                                                                     114
      READ(F3.END=9)M
                                                                                     115
                                                                                     116
      WAITE (PS)
      MM=2+3+M
                                                                                     117
      READ(F3) (FILE38(1).1=1.MM)
                                                                                    118
      WRITE (PS) (FILE38(I) . I=1.MM)
                                                                                     119
      GC TC 8
                                                                                    120
.
      M=0
                                                                                     121
```



```
WALTE (PS)M
                                                                                    122
10
      CONTINUE
                                                                                    123
C
                                                                                    124
C
          READ AND STURE FILE4
                                                                                    125
c
                                                                                    126
      READ(F4) NCYC.TT.NFLAG
                                                                                    127
      WRITE (PS)NCYC.TT.NFLAG
                                                                                    128
      IF (K.NE.6.AND.K.NE.7.AND.K.NE.111GO TO 12
                                                                                    129
      READ(F4) JNACT
                                                                                    130
      WRITE (PS) JNACT
                                                                                    131
      DO 11 1=1.JNACT
                                                                                    132
      READ(F4) COF
                                                                                    133
      BRITE (PS)CDE
                                                                                    134
11
      CENTINUE
                                                                                    135
      CONTINUE
12
                                                                                    136
                                                                                    137
C
c
         READ AND STORE FILES
                                                                                    138
C
                                                                                    139
      REAC(F5) IOPT.NT.PCT.PD
                                                                                    140
      WRITE(PS) IOPT. NT. PCT. PD
                                                                                    141
      IF (K.LT.7)GU TO 15
                                                                                    142
      READ(F5) NOINS
                                                                                    143
      WRITE (PS) NOINS
                                                                                    144
      READ(F5) FILES
                                                                                    145
      WRITE (PS)FILES
                                                                                    146
      IF (K.NE.8.AND.K.NE.12)GO TO 15
                                                                                    147
      READ(F5.END=14)ID.IEDF.(FD(1).X(1).I=1.IEDF)
13
                                                                                    148
                      ID. IEDF. (FD(I).X(I). I=1. IEDF)
                                                                                    149
      BRITE(PS)
      GC TO 13
                                                                                    150
      10=-1
                                                                                    151
14
      WAITE (PS)
                      ID. IEDF. (FD(1).x(1). I=1. IEDF)
                                                                                    152
15
      CONTINUE
                                                                                    153
                                                                                    154
C
C
      READ AND STORE FILES
                                                                                    155
C
                                                                                    156
      IF(K.LT.3)GO TO 17
                                                                                    157
                                                                                    158
      NACT2=2*NACT
      DC 16 I=1.NBREAK
                                                                                    159
      READ(F6) (FILE6(J). J=1.NACT2)
                                                                                    160
      WRITE(PS)(FILE6(J).J=1.NACT2)
                                                                                    161
16
      CONTINUE
                                                                                    162
17
      CONTINUE
                                                                                    163
C
                                                                                    164
C
         READ AND STORE FILET
                                                                                    165
C
                                                                                    166
                                                                                    167
      IF(K.LT.5)GO TO 20
      READ (F7.END=19) LIST
                                                                                    168
18
      WRITE(PS)
                      LIST
                                                                                    169
      GC TO 18
                                                                                    170
19
      LIST(1)=-1
                                                                                    171
      WRITE (PS)LIST
                                                                                    172
20
      CCNTINUE
                                                                                    173
                                                                                    174
C
C
      BRITE TERMINATION INFORMATION
                                                                                    175
C
                                                                                    176
      BRITE(6.113)PS.NCYC. IPROG
                                                                                    177
109
      STOP
                                                                                    178
      FERMAT (2A4)
                                                                                    179
110
      FORMAT(1HO.5%. THE NAME SPECIFIED FOR THE LAST PROGRAM EXECUTED IS
                                                                                    180
111
                                                                                    181
     * INVALID. 1)
      FORMAT(1H1.132(***)/1x.132(***)/*OTHIS IS THE OUTPUT FOR THE SYSTE
                                                                                    182
```

*M INTERRUPTION PROGRAM: SAVEFIL'/1H0.132(***)/1x.132(***))	183
113 FORMAT (1HO.5X. THE OPERATIONAL INFORMATION GENERATED BY THE STATIS	184
*IICAL PERT PROGRAMS HAS BEEN SUCCESSFULLY TRANSFERRED */6x. *TC THE	185
*DATA SET DEFINED AS 1/0-UNIT *.12//6x. THE PROCEDURE WAS INTERRUPT	186
*ED DURING THE *.15.*-TH ITERATION.*/6x.244.* WAS THE LAST PROGRAM	187
EXECUTED.)	188
END	189

THE STATE OF STATE OF

10. READFIL

```
C
      PREGRAM READFIL
                                                                                  1
                                                                                  2
         THIS FROGRAM RETRIEVES THE OPERATIONAL INFORMATION FOR THE
C
         STATISTICAL PERT PROGRAMS FROM THE PERMANENT FILE REFERENCED BY
C
         I/C-UNIT PS AND RESTORES IT TO THE APPROPRIATE TEMPORARY FILES
C
                                                                                  5
C
                                                                                  6
C
                                                                                  7
c
         INPUT INSTRUCTIONS
                                                                                  8
c
         A SINGLE CARD SHOULD BE SUPPLIED AS INPUT TO THIS PROGRAM.
C
                                                                                 10
c
C
         THE CARD SHOULD BEAR THE FOLLOWING:
                                                                                 12
C
              COLS 1- 8: THE NAME OF THE LAST PROGRAM EXECUTED BEFORE
                                                                                 13
C
                            THE INTERRUPT (FORMAT 2A4, LEFT JUSTIFIED)
                                                                                 14
                           O IF THE PROCEDURE WAS INTERRPUTED BEFORE
c
              COL
                                                                                 15
                      10:
C
                               THE FIRST EXECUTION OF SYNTH
                                                                                 16
C
                              OTHERWISE
                                                                                 17
C
              THE REMAINING PARAMETERS SHOULD ONLY BE SUPPLIED IF THE
                                                                                 16
c
              PROCEDURE WAS INTERRUPTED IMMEDIATELY AFTER THE EXECUTION
                                                                                 19
c
              OF SYNTH
                                                                                 20
C
              COLS 11-20: THE VALUE OF TT FCR THE NEXT ITERATION IF
C
                                                                                 22
C
                            DIFFERENT FROM THE VALUE SPECIFIED BY SYNTH
                                                                                 23
C
              CCLS 21-25:
                            1 IF THE VALUES OF ANY CF THE FOLLOWING
                                                                                 24
c
                               PARAMETERS ARE TO BE DIFFERENT FROM THEIR
                                                                                 25
C
                               CURRENT VALUES IN THE NEXT ITERATION:
                                                                                 26
C
                                   ICPT.PCT.PD
                                                                                 27
C
                            O OTHERWISE
                                                                                 28
C
              COLS 26-30:
                           THE VALUE OF
                                          ICPT FOR THE NEXT ITERATION
                                                                                 29
                           THE VALUE OF PCT FOR THE NEXT ITERATION
C
              CCLS 31-40:
                                                                                 30
              COLS 41-50:
                           THE VALUE OF PD FOR THE NEXT ITERATION
C
                                                                                 31
C
                                                                                 32
C
         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS. LET
                                                                                 33
         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL
C
                                                                                 34
C
                     PROJECT NETWORK
                                                                                 35
         BKMAX = THE MAXIMUM NUMBER OF BREAK PCINTS ALLOWED IN THE
C
                                                                                 36
c
                     ENTIRE PROJECT'S TIME-COST CURVE
                                                                                 37
c
         NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR
                                                                                 38
C
                     EACH ACTIVITY
                                                                                 39
c
         IECFMAX = THE MAXIMUM NUMBER OF SUBDIVISIONS ALLOWED IN THE
                                                                                 40
                     APPROXIMATE COF FOR EACH SUBNETWORK
C
                                                                                 41
         NCINSMAX = THE MAXIMUM NUMBER OF INSTRUCTIONS GENERATED BY THE
C
                                                                                 42
C
                     DECOMPOSITION PROCESS
                                                                                 43
C
         CURRENTLY, MMAX=1000; BKMAX=3000; NCTMAX=6; IEDFMAX=20;
                                                                                 44
c
                     NCINSMAX=100
                                                                                 45
C
                                                                                 46
c
      INTEGER#4 FILEO(4*MMAX+5).NCT(MMAX).FILE3B(3*MMAX+5).
                                                                                 47
C
                FILES(NOINSMAX#27)
                                                                                 48
C
      INTEGER+2 BREAK(BKMAX).TIMCST(2+NCTMAX).FILE6(2+MMAX)
                                                                                 49
      REAL * E FD(IEDFMAX) . X(IEDFMAX)
C
                                                                                 50
C
      EGUIVALENCE (FILEO(3+MMAX+6).NCT(1))
                                                                                 51
                                                                                 52
      INTEGER F0/8/.F1/9/.F2/10/.F3/11/.F4/12/.F5/13/.F6/14/.F7/15/
                                                                                 53
      INTEGER PS/4/
                                                                                 54
      INTEGER FILEO(4005) .NCT(1000) .FILE3A(11) .FILE3B(3005) .FILE5(2700).
                                                                                 55
              L15T(24)
                                                                                 56
      REAL FILE1(5) .FILE2(5)
                                                                                 57
      REAL +8 CDF(12).FD(20).X(20)
                                                                                 58
      INTEGER#2 NUREAK, BREAK(3000). TIMCST(12). TEST1. TEST2. TEST3. STIME.
                                                                                 59
                FILE6(2000)
                                                                                 60
```

```
EQUIVALENCE (FILEO(1).NACT).(FILEO(3006).NCT(1))
EQUIVALENCE (FILE1(1).FILE2(1).FILE3A(1).FILE3B(1).FILE5(1).
                                                                                        61
                   FILE6(1).BREAK(1).TIMCST(1).LIST(1).CDF(1).FD(1))
                                                                                        63
      INTEGER IPROG(2)
                                                                                        64
      INTEGER PROG(13)/*MAIN*,*LOOP*,*DPS *,*DSR *,*SIMP*,*MODS*,*DECO*,
                                                                                        65
                         "SUBN" . "SYNT" . "DSR " . "MODS" . "SUBN" . "SYNT"/
                                                                                        66
                                                                                        67
C
      BRITE(6.112)
                                                                                        68
      READ(5.11G) IPROG.NCYC. IFLG.TTT. 110PT.PPCT.PPD
                                                                                        69
      K1=10
                                                                                        70
      IF (NCYC.EG. O) KI=1
                                                                                        71
      DC 1 1=K1.13
                                                                                        72
      IF (IPROG(1).EQ.PROG(1))GO TO 2
                                                                                        73
1
      CENTINUE
                                                                                        74
      BRITE(6.111)
                                                                                        75
      GC TO 109
                                                                                        76
2
      K= I
                                                                                        77
      REBIND FO
                                                                                        78
      REWIND F1
                                                                                        79
      REBIND F2
                                                                                        60
      REMIND FJ
                                                                                        81
      REBING F4
                                                                                        82
      REBINC F5
                                                                                        23
      REWING F6
                                                                                        84
      REWIND F7
                                                                                        85
      REWIND PS
                                                                                        86
                                                                                        87
C
C
         RESTORE FILEO
                                                                                        88
                                                                                        AQ
      REAC(PS) FILEO
                                                                                        90
      WRITE(FO)FILEO
                                                                                        91
      REAC(PS) JMAT
                                                                                        92
      WRITE(FO) JMAT
                                                                                        93
      IF (K.LT.3)GU TO 3
                                                                                        94
      READ(PS) NBREAK. (BREAK(I). I=1. NBREAK)
                                                                                        95
      WAITE (FO) NBREAK, (BREAK(I), 1=1, NBREAK)
                                                                                        96
      ENCFILE FO
                                                                                        97
3
      CCNTINUE
                                                                                        98
C
                                                                                        99
C
          RESTORE FILE!
                                                                                       100
C
                                                                                       101
      REAC(PS) TEST1.TEST2.TEST3.STIME
                                                                                       102
      WAITE (F1) TEST1. TEST2. TEST3.STIME
                                                                                       103
      CC 4 I=1.NACT
                                                                                       104
      AACT=ACT(1)
                                                                                       105
      AACTI=AACT-1
                                                                                       106
      NNCT2=2*NACT
                                                                                       107
      READ(PS) (TIMCST(J).J=1.NNCT2)
                                                                                       108
       BRITE(F1)(TIMCST(J).J=1.NNCT2)
                                                                                       109
                                                                                       110
      DO 4 J=1.NNCT1
      READ(PS) FILE1
                                                                                       111
       WAITE (FI) FILE
                                                                                       112
      CENTINUE
                                                                                       113
       ENOFILE FI
                                                                                       114
                                                                                       115
C
          RESTORE FILE2
                                                                                       117
       IF (K.NE.4.AND.K.NE.5.AND.K.NE.10)GD TO 6
                                                                                       118
      00 5 1=1 . NACT
                                                                                       119
      REAU(PS) FILE2
                                                                                       120
       BAITE(F2)FILE2
                                                                                       121
```

CONTINUE ENCFILE F2	
CENTINUE	
RESTORE FILES	
REAC(PS) FILEJA	
WEITE (F3) FILE3A	
IF (K.NE.S.AND.K.NE.6)GU TO T	
READ(PS) FILE38	
WEITE(F3)FILE38	
CENTINUE	
IF(K.LT.7)GU TO 10	
READ (PS)M	
1F (M.EQ.0) GO TO 9	
WRITE(F3)M	
MM=2+3+M	
READ(FS) (FILE38(I).I=1.MM)	
WAITE(F3)(FILE3B(I), I=1.MM)	
GC TO 8	
CENTINUE	
ENOFILE F3	
CONTINUE	
RESTORE FILE4	
READ(PS) NCYC.TT.NFLAG	
IF (TTT.NE.0.0) TT=TTT	
WRITE(F4)NCYC.TT.NFLAG	
IF (K.NE. 6. AND. K. NE. 7. AND. K.	NE-11)GO TO 12
READ (PS) JNACT	
MRITE(F4) JNACT	
DO 11 1=1.JNACT	
READ(PS) COF	
WRITE(F4)CDF	
CENTINUE	
CONTINUE	
RESTORE FILES	
READ(PS) IOPT.NT.PCT.PD	
IF(IFLG.EG.O)GO TO 21	
10PT=110PT	
PCT=PPCT	
PO=PPD	
BELTE(F5) LOPT . NT . PCT . PD	
IF(K.LT.7)GO TO 15	
READ(PS) NOINS	
WELTE (F5) NOINS	The same of the same of the same of the
HEAD(PS) FILES	12 TO MAY 1 NO 1 TO
WEITE (F5) FILES	
IF (K.NE. B. AND.K.NE. 12)GO TO	
	(),x((),(=1,(EDF)
IF(IC.EQ1)GO TO 14	
	1).X(1),[=1.[EDF]
GU TO 13	
CCNTINLE	
ENCFILE F5	
ENGFILE FS	
ENGFILE FS	

c		163
	1F(K.LT.3)G0 TO 17	184
	NACT2=2*NACT	185
	DG 16 I=1.NBREAK	186
	READ(PS) (FILE6(J), J=1.NACT2)	187
	WRITE(F6)(FILE6(J), J=1, NACT2)	188
16	CONTINUE	189
	ENDFILE F6	190
17	CCNTINUE	191
C		192
C	RESTORE FILE7	193
C		194
	IF(K-LT-5)GO TO 20	195
18	REAC(PS) LIST	196
	IF(LIST(1).EQ1)GO TC 19	197
	WRITE(F7) LIST	198
	GC TG 18	199
19	CONTINUE	200
	ENCFILE F7	201
20	CCNTINUE	202
C		203
C	WRITE TERMINATION INFORMATION	204
C		205
	mRITE(6.113) NCYC. IPROG	206
109	STOP	207
110	FORMAT (244.12.F10.5.215.2F10.5)	208
111	FORMAT(1H0.5X. THE NAME SPECIFIED FOR THE LAST PROGRAM EXECUTED IS	209
	• INVALID.•)	210
112	FORMAT(1H1.432(***)/1x.132(***)/*OTHIS IS THE OUTPUT FOR THE SYSTE	211
	M RESTART PROGRAM: READFIL/1H0.132(***)/1x.132(***))	212
113	FCHMAT(1H0.5X. THE OPERATIONAL INFORMATION GENERATED BY THE STATIS	213
	*TICAL PEHT PROGRAMS HAS BEEN SUCCESSFULLY RESTORED 1/6x. *TC THE	214
	TEMPORARY CATA SETS.. //6x.*THE PROCEDURE WAS INTERRUPT	215
	*ED DURING THE *.15TH ITERATION. */6X.2A4. * WAS THE LAST PROGRAM	216
	EXECUTED..)	217
	END	218

ENTATION OF THE	28. REPORT SEC Und 28. GROUP	classified classified classified
	Und	classified
	26. GROUP	classified
	Unc	
	NEW PROJECT	r SCHEDULING
	NEW PROJECT	r SCHEDULING
	NEW PROJECT	SCREDULING
elken Jr.		
elken Jr.	-	
elken Jr.		
elken Jr.		
elken Jr.		
TZA TOTAL NO O	E PAGES T	7b. NO. OF REFS
	The second secon	12
00.000000000000000000000000000000000000		
Se. ORIGINATOR	TREFORT NUMBE	EH(3)
	Report No	57
	Report No.	
9b. OTHER REPO	RT NO(5) (Any other	er numbers that may be assigned
oublic release a	nd sale; it	s distribution
12. SPONSORING	MILITARY ACTIVI	TY
(T+1)		
Mark Williams Inches		
	9b. OTHER REPO	301 90. ORIGINATOR'S REPORT NUMBER Report No

This report documents a new project scheduling procedure developed at the which Institute of Statistics, Texas A&M University. The project scheduling algorithm, is a five step iterative procedure capable of determining a minimum cost project schedule when the activities making up the project have durations which are random variables. The cost of an activity is assumed to be a convex piecewise linear function of the activity's mean duration. The problem is to determine the activity mean durations which both minimize the total project cost and insure that the mean (or some specified percentile) of the corresponding project completion time distribution is less than or equal to a specified project deadline.) The entire distribution of the project's completion time under the minimum cost schedule is a valuable by-product. Information on the trade-off between the project's minimum cost and its specified deadline is also provided.

This report is intended as a user's guide to the new project scheduling procedure and its computer implementation. The report includes a description of the project scheduling problem, a general overview of the scheduling procedure including references to technical reports documenting the development of the procedure, and an example of the procedure's performance. The documentation of the computer implementation includes specific input instructions; sample input and output; flowcharts; individual program descriptions; technical details concerning temporary data sets, job control language, and program interruption and restart procedures; and program listings.

ATTACHMENT III

חח	FORM	1	A	7	2
טט	1 NOV 65	ı	4	1	J

Unclassified
Security Classification LINK A LINK . LINK C KEY WORDS ROLE ROLE ROLE WT Statistical PERT Project Scheduling Minimum Cost Scheduling Random Activity Durations Project Completion Time Distribution Network Duration Distribution User's Guide Computer Programs for PERT

ATTACHMENT III (Continued) DD PORM 1473 (BACK)

Unclassified

BASIC DISTRIBUTION LIST FOR UNCLASSIFIED TECHNICAL REPORTS

Operations Research Office of Naval Research	Copies	Director Office of Naval Research	Copies
(Code 434)		Branch Office	
Arlington, Virginia 22217	3	495 Summer Street	
maringeon, trapana arms		Boston, Massachusetts 02210	
Director, Naval Research Laboratory		Attn: Dr. A.L. Powell	1
Attn: Library, Code 2029 (ONRL)			
Washington, D.C. 20390	2	Prof. Emanuel Parzen	
washington, D.C. 20070		Institute of Statistics	
Defense Documentation Center		Texas A&M University	
Cameron Station		College Station, TX 77843	
Alexandria, Virginia 22314	12		1
Defense Logistics Studies		Office of Naval Research	
Information Exchange		San Francisco Area Office	
Army Logistics Management Center		760 Market St Room 447	
Fort Lee, Virginia 23801	1	San Francisco, California 94103	1
Technical Information Center		Technical Library	
Naval Research Laboratory		Naval Ordnance Station	
Washington, D.C. 20390	6	Indian Head, Maryland 20640	1
Office of Naval Research		Bureau of Naval Personnel	
New York Area Office		Navy Department	
207 West 24th Street		Technical Library	
New York, New York 10011		Washington, D.C. 20370	
Attn: Dr. J. Laderman	1	STOP 82	2
Director, Office of Naval Research	h	Library, Code 0212	
Branch Office		Naval Postgraduate School	
1030 East Green Street		Monterey, California 93940	1
Pasadena, California 91101			
Attn: Dr. A.R. Laufer	1	Naval Ordnance Station	
Yale University		Louisville, Kentucky 40214	1
Department of Statistics		Library	
Box 2179 - Yale Station		Naval Electronics Laboratory	
New Haven, CT 06520	1	Center	
Attn: Prof. I.R. Savage		San Diego, California 92152	1
Mr. William Dejka			
Code 360		Naval Ship Engineering Center	
Naval Electronics Laboratory Cent	er	Philadelphia	
San Diego, California 92132	1	Division Technical Library	
		Philadelphia, Pennsylvania 19112	1
Professor Lucien A. Schmit, Jr.			
Room 6731, Boelter Hall		Dr. A.L. Slafkosky	
School of Engrag. & Applied Sci.		Scientific Advisor	
University of California		Commandant Marine Corps (Code AX)	
Los Angeles, California 90024	1	Washington, D.C. 20380	1

Purdue University	Copies	Dr. Del Cilmer	Copies
Graduate School of Ind. Admin.	•	Naval Weapons Center (Code 607)	
Lafayette, Indiana 47907		China Lake, California 93555	1
Attn: Prof. Andrew Whinston	1		
Department of Economics		Dr. Paul Murrill	
Tisch Hall - 5th Floor		Project Themis	
Washington Square, NY Univ.		Department of Chemical Engineering	g
New York City, NY 10003		Louisiana State University	
Attn: Prof. O. Morgenstern	1	Baton Rouge, Louisiana 70803	1
Dept. of Statistics		Director	
Syntex Research		Army Materials & Mechanics Research	ch
3401 Hillview		Center	
Palo Alto, CA 94304	1	Attn: Mr. J. Bluhm	
Attn: Prof. Stuart Bessler		Watertown, Massachusetts 02172	1
University of California			
Department of Engineering		Commanding Officer	
Los Angeles, California 90024		U.S. Army Ballistic Research	•
Attn: Frof. R.R. O'Neill	1	Laboratories	
		Attn: Dr. John H. Giese	
Maritime Transportation Research		Aberdeen Proving Ground,	
Board		Maryland 21005	1
National Academy of Sciences			
2101 Constitution Avenue		University of Iowa	
Mashington, D.C. 20418		Department of Mechanics	
Attn: RADM J.B. Oren USCG (RET)	1	and Hydraulics	
		Iowa City, Iowa 52240	
Stanford University		Attn: Prof. W.F. Ames	1
Department of Operations Research			
Stanford, California 94305	in the later	Office of Naval Research	1
Attn: Prof. A.F. Veinott, Jr.	1	Resident Representative	
		1105 Guadalupe	
Texas A&M Foundation		Lowich Building	
College Station, Texas 77843		Austin, Texas 78701 Attn: Mr. Francis M. Lucas	
Attn: Prof. H.O. Hartley	one mand	Attn: Mr. Francis M. Lucas	1 .
Case Western Reserve University		Dr. Jerome Bracken	
Cleveland, Ohio 44106		Institute of Defense Analyses	
Attn: Prof. B.V. Dean	1	400 Army-Navy Drive	
		Arlington, Virginia 22202	1
University of California			
Center for Research in		Professor Richard L. Fox	
Management Science		School of Engineering	
Berkeley, California 94720		Case Western Reserve University	
Attn: Prof. W.I. Zangwill	1	Cleveland, Ohio 44106	1
U.S. Naval Postgraduate School			
Department of Operations		The second secon	a Lament
Research and Economics			1 11
Monterey, California 93940			
Attn: Prof. C.R. Jones			

	Copies	Harvard University	Copies
		Department of Statistics	
		Cambridge, Massachusetts 02139	
		Attn: Prof. W.G. Cochran	1
	1		
		Columbia University	
		Department of Industrial	
		Engineering	
		New York, New York 10027	
	1	Attn: Prof. C. Derman	1
University of Chicago		New York University	
Statistical Research Center		Institute of Mathematical	
Chicago, Illinois 60637		Sciences	
Attn: Prof. W. Kruskal	1	New York, New York 10453	
Acti. Flot. W. Ridskar		Attn: Prof. W.M. Hirsch	1
Stanford University		Acti, 1101. Will Hillon	
Department of Statistics		University of North Carolina	
Stanford, California 94305		Statistics Department	
Attn: Prof. G.J. Lieberman	1	Chapel Hill, North Carolina 27515	
Attn: Plot. G.J. Lieberman	and the second second	Attn: Prof. W.L. Smith and	
		Prof. M.R. Leadbetter	1
		PIOI. M.R. Leadbetter	
		Purdue University	
	1	Division of Mathematical Sciences	
		Lafayette, Indiana 47079	
Princeton University		Attn: Prof. H. Rubin	1
Department of Mathematics			
Princeton, New Jersey 08540		University of California, San Dieg	20
Attn: Prof. J.W. Tukey	1	Department of Mathematics	
Accii. Troc. ora. rakey		P.O. Box 109	
Stanford University		La Jolla, California 92038	
Department of Statistics		Attn: Prof. M. Rosenblatt	1
Stanford, California 94305		Acti. Hor. in Rosenstate	
Attn: Prof. T.W. Anderson	1	Florida State University	
Attn: Flot. 1.w. Anderson	to and the last	Department of Statistics	
W-1			
University of California		Tallahassee, Florida 32306	1
Department of Statistics		Attn: Prof. R.A. Bradley	
Berkeley, California 94720	You bear and	War Wart Water and the	
Attn: Prof. P.J. Bickel	1	New York University	
		Department of Industrial	
University of Washington		Engineering & Operations	
Department of Mathematics		Research	
Seattle, Washington 98105	mod he the	Bronx, New York 10453	
Attn: Prof. Z.W. Birnbaum	formula state of	Attn: Prof. J.H.K. Kao	1
Cornell University		University of Wisconsin	
Department of Mathematics		Department of Statistics	
White Hall		Madison, Wisconsin 53706	
Ithaca, New York 14850		Attn: Prof. G.E.P. Box	1
Attn: Proi J. Kiefer	1		
Accii. 1.02. J. Kletel			

Logistics Research Project	Copies	The University of Michigan	Copies
The George Washington University		Department of Mathematics, M.E.	
707 - 22nd Street, N.W.		Ann Arbor, Michigan 48104	
Washington, D. C. 20037		Attn: Prof. R.M. Thrall	1
Attn: Dr. W. H. Marlow	1		
		Princeton University	
International Business Machines		Department of Mathematics	
Corporation		Princeton, New Jersey 08540	
P.O. Box 218, Lamb Estate		Attn: "rof. A.W. Tucker	1
Yorktown Heights, New York 10598		O Washing Dansens Hadronolty	
Attn: Dr. Alan Hoffman	1	Case Western Reserve University	
University of California		Systems Research Center	
University of California Management Sciences Research		Cleveland, Ohio 44106 Attn: Prof. M. Mesarovic	1
Project		Attn: Prof. M. Mesalovic	
Los Angeles, California 90024	•	University of Texas	
Attn: Dr. J.R. Jackson	1	Department of Mathematics	
Feel, Dr. J.R. Jackson		Austin, Texas 78712	
Harvard University		Attn: Dr. A. Charnes	1
Department of Economics	,	Accii. Dr. II. diazza	
Cambridge, Massachusetts 02138		Stanford University	
Attn: Prof. K.J. Arrow	1	Department of Operations Research	
		Stanford, California 94305	
Cowles Commission for Research		Attn: Dr. D.L. Iglehart	1
in Economics			
Yale University		University of Delaware	
New Haven, Connecticut 06520		Department of Mathematics	
Attn: Prof. Martin Shubik	1	Newark, Delaware 19711	
		Attn: Prof. R. Remage, Jr.	1
Carnegie-Mellon University			
Graduate School of Industrial		Stanford University	
Administration		Department of Operations Research	
Pittsburgh, Pennsylvania 15213		Stanford, California 94305	
Attn: Prof. G. Thompson	1	Attn: Prof. F.S. Hillier	1
		Dr. Claude-Alain Burdet	
University of California			
Department of Economics		Asst. Prof. Industrial Admin. Carnegic-Mollon University	
Berkeley, California 94720 Attn: Prof. R. Radner	,	Pittsburgh, Pennsylvania 15213	1
	all week	ricesouren, remisgivania iyeis	
University of California		Stanford University	
Operations Research Center		Department of Operations Research	
Institute of Engineering Research		Stanford, California 94305	
Berkeley, California 94720		Attn: Prof. G. B. Dantzig	1
Attn: Prof. D. Gale	1	neem, received as a sense of	
need: 11017 bt 3010	- T	. Chief of Naval Research (Code 436)	
University of California		Department of the Navy	
Graduate School of Business		Arlington, Va. 22217	1
Administration			
Los Angeles, California 90024		Science Librarian	
Attn: Prof. J. Marschak	1	Kresge Library	
		Oakland University	
		Rochester, Michigan 48063	1

University of Connecticut Department of Statistics	Copies	Professor Geoffrey 3. Watson, Chairman	Copies
Storrs, Connecticut 06268		Department of Statistics	
Attn: Prof. H.O. Posten	1	Princeton University	
		Fine Hall	•
		Princeton, N. J. 08540	1%
		Stanford University	
		Department of Mathematics	
	1	Stanford, California 94305	
		Attn: Prof. S. Karlin'	1
ARCON Corporation			
Lakeside Office Park		University of Sheffield	
North Avenue at Route 128		Department of Probability	
Wakefield, Massachusetts 01880		and Statistics	
Attn: Dr. A. Albert	1	Sheffield 10, ENGLAND	
		Attn: Prof. J. Gani	1
Stanford University			
Department of Statistics		University of California	
Stanford, California 94305		Operations Research Center	
Attn: Prof. H. Chernoff	1	Institute of Engineering Research	
		Berkeley, California 94720	
Yale University		Attn: Prof. R.E. Barlow	1
Department of Statistics			
New Haven, Connecticut 06520		Stanford University	
	1	Department of Statistics	
		Stanford, California 94305	
Rutgers-The State University		Attn: Prof. H. Jolomon	1
Statistics Center			
New Brunswick, New Jersey 08903		Applied Mathematics Laboratory	
Attn: Prof. H.F. Dodge	1	Naval Ships Research Development Center	
Yale University		Washington, D.C. 20007	1
Department of Statistics			
New Haven, Connecticut 06520		Systems Analysis Division	
Attn: Prof. F. J. Anscombe	1	Room BE760, Pentagon	
		Washington, D.C. 20350	
Purdue University		Attn: Mr. A.S. Rhodes, Op-964	1
Division of Mathematical Jciences	2010000		
Lafayette, Indiana 47907		Department of Statistics	
Attn: Prof. 3.3. Gupta	1	University of North Carolina	
		Chapel Hill, North Carolina 27515	
Cornell University		Attn: Prof. M.R Leadbetter	1
Department of Industrial			
Engineering		Southern Methodist University	
Ithaca, New York 19850		Department of Statistics	
Attn: Prof. R.E. Bechhofer	1	Dullas, T xas 75222	
The state of the s		Attn: Prof. D.B. Owen	1
Mrs. Barbara Eaudi			
Univ. Program Coordinator, B.E.		Israel Institute of Technology	
NASA Johnson Space Center		Technion	
Houston, TX 77058	1	Haifa, ISRAEL	
		Attn: Prof. P. Naor	

.

Office of Gifts & Exchanges	Copies
Library Texas A&M University College Station, Texas 77843	2
Archives Texas A&M University	
College Station, TX 77843	. 1